



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TREBALL FI DE GRAU

Grau en Enginyeria Electrònica Industrial i Automàtica

SISTEMA GLOBAL DE LOCALITZACIÓ GPS

VIA DISPOSITIU MÒBIL



Memòria

Autor:	Vicenç Suriol Luján
Director:	Joan Segura Casanovas
Convocatòria:	Juny 2017

Resum

L'Internet de les coses comença a ser una realitat en un mercat tecnològic que avança a velocitats exponencials. Cada vegada apareixen més dispositius autònoms connectats a la xarxa capaços de facilitar-nos el dia a dia i tot apunta que seguirà així en els pròxims anys.

En aquest projecte s'ha dissenyat i creat un prototip capaç de rebre coordenades GPS i enviar-les mitjançant la xarxa de telecomunicacions 2G (molt emprada per a comunicació M2M) a un servidor programat per a què les guardi a una base de dades i les faciliti a una aplicació mòbil per a poder observar-ho en un mapa també creada en la seva totalitat per a satisfer les necessitats del projecte.

Trobareu doncs en aquesta memòria tot el procediment, medis, components electrònics i software que s'ha requerit per a la creació completa del projecte i una explicació concisa, però sobretot entenedora, de cada apartat.

El treball conclou amb uns resultats satisfactoris i amb un seguit d'ampliacions per a tot el conjunt en general ja que la facilitat per a expandir el projecte i per a afegir-hi noves funcionalitats ha resultat ser sorprenentment elevat.

Resumen

El internet de las cosas empieza a ser una realidad en un mercado tecnológico que avanza a velocidades exponenciales. Cada vez aparecen más dispositivos autónomos conectados a la red capaces de facilitarnos el día a día y todo apunta a que seguirá siendo así en los próximos años.

En este proyecto se ha diseñado y creado un prototipo capaz de recibir coordenadas GPS y enviarlas mediante la red de telecomunicaciones 2G (muy usada para comunicación M2M) a un servidor programado para que las guarde en una base de datos y las facilite a una aplicación móvil para poder observarlo en un mapa también creada en su totalidad para satisfacer las necesidades del proyecto.

Encontraréis en esta memoria todo el procedimiento, medios, componentes electrónicos y software que se ha requerido para la creación completa del proyecto y una explicación concisa, aunque por encima de todo, entendedora, de cada apartado.

El trabajo concluye con unos resultados satisfactorios y un conjunto de ampliaciones para todo el conjunto en general ya que la facilidad para expandir el proyecto e incluir nuevas funcionalidades ha resultado ser sorprendentemente elevado.

Abstract

Internet of things starts to be a reality in a technologic market which progresses at exponential speeds. More and more autonomous devices connected to the network are created these days aiming on making our life easier, and everything is expected to keep on this line for the next upcoming years.

In this project we have designed and created a prototype capable of receiving GPS coordinates and sending them through the 2G network telecommunications (widely used for M2M communication), to a web server, programmed to store in a database the data that then will be provided to a smartphone application also created for satisfying the needs of the project.

You will find in this document all the procedure, media, electronics and software that have been required for the complete creation of the project, and a concise but understandable explanation about each section.

The study concludes with satisfactory results and also with some extensions that have been added because of the easiness to expand the project, and adding new functionalities has proven to be surprisingly high.



Agraïments

M'agradaria agrair primer tot el suport que he rebut a casa per part de la meva família tots aquests anys, per a transmetre'm una educació que m'ha animat a voler seguir estudiant i aprendre; Valors dels quals avui en dia admiro i estic molt orgullós. Sense els meus pares estic segur que no seria qui sóc avui en dia.

Als meus amics, que sempre estan per a distreure'm i ajudar-me a la seva manera i entre tots sembla que ens animem a seguir treballant i estudiant; Això realment crea un ambient molt agradable. També a la meva parella, l'Anna qui ha canviat la meva manera de veure les coses i m'ha donat suport durant tot el meu pas per la universitat.

Pel que fa al projecte, vull donar-li les gràcies al meu professor de programació Samir Kanaan, sense la seva ajuda no hagués sigut possible utilitzar un servidor i la feina de l'aplicació s'hagués complicat en gran mesura. També al meu tiet, Jaume Puig per a ajudar-me a tenir les idees clares. De fet sense la seva opinió, no m'hagués interessat per la oferta d'aquest treball de GPS.

Per últim al meu tutor, Joan Segura per a resoldre'm els dubtes que no han estat pocs, ajudar-me quan ho he necessitat i per a permetre'm realitzar un projecte del qual he après molt i considero el millor treball dels meus estudis superiors tant per la dedicació i les hores invertides en ell, com per la funcionalitat i la professionalitat amb la qual s'han abastat tots els temes d'aquest.



Glossari

1G a 5G	Primera a Cinquena Generació de tecnologies mòbils.
3GPP	3rd Generation Partnership Project (Projecte Associació de Tercera Generació).
API	Application Programming Interface (Interfície de programació d'aplicacions).
APN	Access Point Name (Nom del Punt d'Accés).
AT Commands	Llenguatge de comandes d'estàndard obert per a configurar Mòdems .
AT&T	American Telephone and Telegraph, Inc (Companyia de telecomunicacions).
Baud Rate	Taxa de bauds o baudis. Nombre de unitats de senyal per segon.
C / N₀	Carrier-to-Noise Ratio (relació senyal/soroll d'una senyal).
Char	Tipus de variable en programació que contenen un caràcter en 1 byte.
Dataclip	Permeten mostrar consultes a una Base de dades allotjada a un servidor.
dB	Decibel. Unitat relativa d'una senyal elèctrica que expressa una raó entre quantitats.
Direcció IP	Identificador d'un dispositiu que utilitzi el Protocol IP per a entrar a la xarxa.
EDGE	Enhanced Data rates for GSM of Evolution (Taxes de Dades Millorades per GSM)
Float	Tipus de variable en programació que conté un valor numèric decimal.
FTDI	Future Technology Devices International. Empresa especialitzada en tecnologia USB.
GPRS	General Packet Radio Service (Servei General de Paquets via Ràdio).
GPS	Global Positioning System (Sistema de posicionament Global).
GSM	Global System for Mobile (Sistema Global per les comunicacions Mòbils).
H, H+, HSDPA	High Speed Downlink Packet Access (Xarxa sense fils entre la 3a i 4a generació).
HTTP/HTTPS	Hipertext transfer protocol / Hipertext transfer protocol secure.
IC	Integrated Circuit (Circuit Integrat).
IDE	Integrated Development Environment (Entorn de desenvolupament integrat).
IoT	Internet of Things. Terme per a la interconnexió d'elements quotidians amb internet.
ITU	International Telecommunications Union (Unió internacional de Telecomunicacions).
JSON	JavaScript Object Notation. Representa un format de text d'intercanvi de dades.
M2M	Machine to Machine communication (Comunicació entre màquines).
MMS	Multimedia Message Service (Servei de Missatges Multimèdia).
Mòdem	Modulador-Desmodulador per a transmetre senyals per cable o ones.
NDK	Native Development Kit (Kit de desenvolupament Natiu).
NMEA	National Marine Electronics Association. Especificació que utilitzen els receptors GPS.
PRN	Pseudorandom noise (Soroll Pseudoaleatori).
SAW filter	Surface Acoustic Wave Filter. Filtre electromecànic utilitzat en radiofreqüència.
SDK	Software Development Kit (Kit de desenvolupament de Software).
SMS	Short Message Service (Servei de missatges Curt).
String	Tipus de variable en programació que conté un conjunt de caràcters.
TCP/IP	Conjunt de protocols per a la comunicació entre nodes de la xarxa.
TTF	Time To First Fix (temps a fixar per primer cop).
UART	Universal Asynchronous Receiver-Transmitter (R-T Asíncron Universal).
UMTS	Universal Mobile Telecommunications System. Tecnologia de 3a generació mòbil.
WAN	Wireless Area Network (Xarxa d'àrea sense fils. WLAN WPAN...).
WiMax	Worldwide Interoperability for Microwave Access (Interoperabilitat mundial d'accés per microones).



Índex

RESUM	III
RESUMEN	IV
ABSTRACT	V
AGRAÏMENTS	VII
GLOSSARI	IX
ÍNDEX DE FIGURES	XV
1. PREFACI	1
1.1. Origen del treball	1
1.2. Motivació	1
1.3. Requeriments previs	2
2. INTRODUCCIÓ	5
2.1. Objectius del treball.....	5
2.2. Abast del treball.....	6
3. FONAMENTS	9
3.1. Global Positioning System	9
3.1.1. Característiques tècniques del sistema.....	9
3.1.2. Altres sistemes de navegació: GLONASS, Beidou i Galileo	9
3.1.3. Funcionament del sistema i del receptor GPS	10
3.1.4. L'antena	13
3.2. Transmissió de dades.....	15
3.2.1. Xarxes sense fils de poc abast	15
3.2.2. Xarxes sense fils de llarg abast	16
3.2.3. Modulació de les ones de ràdio	17
3.2.4. Desmodulació	20
3.2.5. Tecnologies i generacions de les xarxes de comunicacions mòbils.....	22
3.3. El Microprocessador	25
3.4. Emmagatzematge de dades	26
3.4.1. El servidor	28
3.4.2. Família de protocols d'internet	29

3.5.	Introducció a programació de mòbils	33
3.5.1.	Les plataformes mòbils	33
3.5.2.	Requisits previs i material necessari.....	37
3.5.3.	Exemples i aplicacions de Qt Creator	40
4.	ESTUDI DE MERCAT	49
4.1.	Anàlisi del consumidor	49
4.2.	Anàlisi del mercat actual	50
5.	ESTAT DE L'ART	53
5.1.	Receptor GPS	53
5.1.1.	L'avanç de la tecnologia: DGPS i RTK.....	55
5.2.	Cobertura de transmissió de dades	56
5.2.1.	Carrier Agregation i el suposat 4G+	57
5.2.2.	El projecte AirGig	58
5.3.	Servidors	59
5.4.	Aplicacions de plataforma mòbil.....	61
5.4.1.	Android, iOS, els dos?	61
5.4.2.	Programa per a la creació de la app	63
6.	SELECCIÓ DEL MATERIAL	65
6.1.	Mòdul receptor GPS	65
6.1.1.	Models de mercat GPS.....	65
6.1.2.	Elecció del receptor GPS	68
6.2.	Transmissor de dades	69
6.2.1.	Requeriments del mòdul: 2G o 3G?	69
6.2.2.	Models de mercat 2G	71
6.2.3.	Models de mercat 3G	73
6.2.4.	Elecció del mòdul	73
6.2.5.	Targeta SIM	74
6.3.	Alimentació	76
6.4.	Altres	77
7.	PLANIFICACIÓ DE LES TASQUES	79
8.	POSADA EN MARXA	81
8.1.	Receptor GPS	81
8.1.1.	Hardware del mòdul GPS.....	82
8.1.2.	Comprovació del correcte funcionament del receptor GPS	84

8.1.3.	Disseny inicial del software GPS.....	89
8.2.	Mòdul SIM900.....	91
8.2.1.	Hardware del mòdul SIM900	92
8.2.2.	Comprovació i primeres proves del mòdul SIM900	93
8.3.	Circuit d'alimentació i càrrega	104
9.	CREACIÓ DEL SERVIDOR.....	107
9.1.	CLI Toolbelt i fitxers pel servidor de Heroku	107
9.2.	La Base de dades.....	108
9.2.1.	Programació de la Base de Dades	109
9.3.	Programació del Servidor.....	110
10.	DISSENY DEL CODI D'ARDUINO	115
10.1.	Software del GPS.....	116
10.2.	Software del GPRS	119
10.2.1.	Freqüència d'actualització de dades i el mode Sleep	121
10.3.	Unió i disseny d'un codi operatiu	124
10.4.	Robustesa del codi	127
10.4.1.	Informació incorrecte del GPS.....	127
10.4.2.	Reiniciar per software el receptor GPS	128
10.4.3.	Nombre de satèl·lits	129
10.4.4.	Coordenades negatives	130
11.	CREACIÓ DE L'APLICACIÓ MÒBIL.....	131
11.1.	Què ha de fer l'aplicació? Objectius inicials.	131
11.2.	Comparativa amb una aplicació real	133
11.3.	Disseny inicial	135
11.3.1.	Esquema bàsic i interaccions entre els elements	135
11.3.2.	Disseny de les finestres de l'aplicació	136
11.3.3.	Prova de l'app a un dispositiu mòbil	147
11.4.	Programació de l'aplicació	149
11.4.1.	Botons de canvi de pantalles.....	149
11.4.2.	Connexió amb el servidor.....	150
11.5.	Ampliació de l'aplicació.....	152
11.5.1.	Animacions	152
11.5.2.	Distància entre dos punts.....	153
11.5.3.	Nombre de satèl·lits	154
11.5.4.	Velocitat entre dos punts	155

11.5.5. Simulació de la posició actual	156
11.5.6. Localització GPS del mòbil	159
11.5.7. Buscador lliure de posició	160
11.5.8. Alternar mapes.....	161
12. ANÀLISI DE L'IMPACTE AMBIENTAL	165
CONCLUSIONS	167
PRESSUPOST	169
Costos d'enginyeria	169
Mà d'obra	170
Materials	170
Cost Total	171
OPINIÓ, VALORACIÓ DEL PROJECTE	173
BIBLIOGRAFIA	177
Referències bibliogràfiques	177
Bibliografia complementària	183

Índex de figures

Figura 2.1. Gràfic amb l'objecte del projecte.	6
Figura 3.1. Figura amb la posició de diferents constel·lacions de satèl·lits. (Font: Satellite Navigation [2]).....	10
Figura 3.2. Figura de localització amb tres satèl·lits. (Font: David Pérez [4])	11
Figura 3.3. Figura sobre la relativitat afectada pels satèl·lits. (Font: Write Science [7])	12
Figura 3.4. Proposta d'esquema elèctric d'una antena activa. (Font: Electronic design [10])	15
Figura 3.5. Formats Patch i Hèlix d'antenes passives . (Font: U Blox [9])	15
Figura 3.6. Esquemàtic amb les diferents tecnologies i marques de xarxes sense fils.....	16
Figura 3.7. Esquemàtic on es contemplen les tres xarxes descrites. (Font: Redes Inalambricas TESCI ISC [12])....	17
Figura 3.8. Esquema amb les diferents modulacions analògiques. (Font: WIKI BOOKS [16])	19
Figura 3.9. Esquema amb diferents modulacions digitals. . (Font: WIKI BOOKS [16]).....	20
Figura 3.10. Circuit i senyal d'entrada i sortida d'un desmodulador d'amplitud. (Font: edumedia [17])	21
Figura 3.12. Diagrama de blocs per a un desmodulador d'angle amb PLL. (Font: profesores.edu [18])	21
Figura 3.14. Esquema de les diferents generacions i el seu abast. (Font: Expansion [23]).....	25
Figura 3.15. Un dels magatzems de centre de dades de Google. (Font: Google, Inc. [25]).....	29
Figura 3.16. Capes del model d'internet i els protocols més importants. (Font: Wikipedia[27]).....	31
Figura 3.17. Gràfica comparativa de mòbils i el seu sistema operatiu.. (Font: Statista [32]).....	34
Figura 3.18. Imatge amb diferents dispositius utilitzant la versió 5.0 d'Android (Font: Android [33]).....	36
Figura 3.19. Interfície del programa Qt Creator.....	38
Figura 3.20. Finestra de descarregar dels components SDK d'Android.	39
Figura 3.21. Consola i finestra del programa realitzat.....	42
Figura 3.22. Finestra del programa 3.	44
Figura 3.23. Informació en API del temps a Barcelona.	46
Figura 3.24. Finestra del programa i informació rebuda del servidor.	47
Figura 4.1. Relotge amb GPS i GSM (Font: Amazon [35]).	51
Figura 4.2. Dispositiu de rastrejament per a persones (Font: Snowfox [36]).....	52
Figura 4.3. Dispositiu de rastrejament T8 (Font: Amazon [37]).	52
Figura 5.1. Imatge de les dades rebudes per a un receptor GPS. (Font: spark fun electronics [38])	54
Figura 5.3. Esquema de funcionament del GPS diferencial. (Font: Dronepic [39])	55
Figura 5.5. Imatge referent a la suma de portadores. (Font: quelcom [43])	57
Figura 5.6. Mapa de cobertura 4G a Espanya per part de Vodafone. (Font: Vodafone España [45])	58
Figura 5.7. Fotografia del vídeo promocional de AirGig. (Font: AT&T [46])	59
Figura 5.8. Pla de pagament dels servidors Heroku. (Font: Heroku[50]).....	60
Figura 5.9. Logotip d'Android i d'Apple simulant divisió d'opinions. (Font: worldversus [51]).....	62
Figura 5.10. Divertida imatge d'Android i Apple. (Font: theandroid-mania [54])	63
Figura 5.11. Plataformes suportades per Unity en la creació de videojocs. (Font: Unity [58]).....	64
Figura 6.1. Imatge del receptor SkyLab SKM53. (Font: Punto Flotante S.A. [59])	66
Figura 6.2. Imatge del component extreta de la pàgina oficial de la companyia. (Font: u-blox [60]).....	66

Figura 6.3. Mòdul GPS en qüestió. (Font: cooking hacks [61])	67
Figura 6.4. Mòdul GPS Adafruit. (Font: Adafruit [62])	67
Figura 6.5. Mòdul LE910 4G+GPS i antenes. (Font: cooking hacks [61])	68
Figura 6.6. Mòdul amb el receptor NEO-6M i l'antena adquirits.	68
Figura 6.7. Mòdul SIM900 de venta online.	71
Figura 6.8. Mòdul SIM900 de cooking hacks. (Font: cooking hacks [61])	72
Figura 6.9. Mòdul 2G d'arduino. (Font: Arduino [63])	72
Figura 6.10. Mòdul 3G GSM GPS de adafruit. (Font: Cetronic [64])	73
Figura 6.11. Mòdul GPRS adquirit.	74
Figura 6.12. Tarifes a escollir i pantalla client de Symio amb el consum i saldo disponible.	75
Figura 6.13. Bateria escollida. (Font: asturinformatica, ebay)	76
Figura 6.14. Mòdul càrrega bateries. (Font: ebay)	77
Figura 7.1. Diagrama de Gaant.	80
Figura 8.1. Esquema elèctric del mòdul NEO-6M. (Font: cnx-software [66])	83
Figura 8.2. Monitor Sèrie d'Arduino amb la connexió del mòdul GPS al port de comunicacions.	85
Figura 8.3. Imatge de Google Maps amb les coordenades rebudes pel receptor GPS.	86
Figura 8.4. Imatge de Google Maps amb les coordenades rebudes pel receptor GPS ampliat.	86
Figura 8.5. Imatge del mòdul FTDI juntament amb el mòdul i un connector mini USB a USB.	87
Figura 8.6. Instantània del programa U-CENTER amb els satèl·lits.	87
Figura 8.7. Instantània del programa U-CENTER amb la posició.	88
Figura 8.8. Imatge del monitor sèrie d'Arduino amb el codi anterior.	91
Figura 8.9. Imatge del programa Tera Term al inicialitzar.	93
Figura 8.10. Resultat de diverses operacions amb el mòdul SIM900.	95
Figura 8.11. Fotografia captada en el moment de la trucada.	97
Figura 8.12. Instantània captada des del mòbil receptor del SMS enviat pel mòdul SIM900.	98
Figura 8.13. Servidor de prova creat per a realitzar el programa.	100
Figura 8.14. Monitor Sèrie amb les dades rebudes.	103
Figura 8.15. Esquema electrònic del circuit d'alimentació.	104
Figura 8.16. Placa amb els components soldats.	105
Figura 9.1. Model entitat-relació de la BD	109
Figura 9.2. Finestra de comandes amb les instruccions descrites	113
Figura 9.3. Finestra d'un navegador amb la BBDD.	114
Figura 9.4. Finestra d'un navegador amb la URL que fa servir l'aplicació.	114
Figura 10.1. Diagrama de flux del codi d'Arduino.	115
Figura 10.2. Imatges del monitor amb un retard de 0ms i un de 600ms.	118
Figura 10.3. Pàgina de Simyo amb el consum del mes d'abril.	122
Figura 10.4. Monitor sèrie amb l'execució del codi desenvolupat.	126
Figura 10.5. Pàgina amb les dades de la BD.	127
Figura 11.1. Diagrama de flux que es seguirà per a programar el codi de l'aplicació.	133
Figura 11.2. Pantalla principal i d'estadístiques de l'aplicació referent (Font: Ilya Bogdanovich).	134
Figura 11.3. Dibuix explicatiu de les pantalles de l'aplicació a dissenyar.	136

<i>Figura 11.4. Disseny dels botons.</i>	138
<i>Figura 11.5. Logotip de l'aplicació.</i>	141
<i>Figura 11.6. Pantalla inicial de l'aplicació.</i>	141
<i>Figura 11.7. Pop ups d'entrada.</i>	142
<i>Figura 11.8. Finestra principal.</i>	145
<i>Figura 11.9. Finestra d'estadístiques.</i>	146
<i>Figura 11.10. Finestra de configuració.</i>	147
<i>Figura 11.11. Aplicació en un dispositiu mòbil HD de 5 polzades.</i>	148
<i>Figura 11.12. Mapa amb esfera de simulació engrandint-se.</i>	158
<i>Figura 11.13. Opció de buscador lliure.</i>	160
<i>Figura 11.14. Dades mal introduïdes.</i>	161
<i>Figura 11.15. Error dels servidors de OSM.</i>	162
<i>Figura 11.16. Diferents tipus de mapes.</i>	163
<i>Figura C.1. Resultat final del prototip.</i>	168

1. Prefaci

En aquest primer capítol es presenta el projecte. S'explicarà el perquè s'ha decidit elegir aquest títol, quina era la idea inicial i les motivacions que van fer decantar-me per escollir-lo.

1.1. Origen del treball

Sempre he pensat que el treball final de grau havia de culminar els meus estudis d'enginyeria per després enfocar-me al món laboral, i és per això que sempre he volgut realitzar-ne un on acabés de profunditzar d'una manera més pràctica que teòrica tots els grans blocs de la meva carrera. A més a més és una realitat que l'electrònica i l'automatització són cada cop més presents al dia a dia i penso que no és cap pèrdua de temps dissenyar algun aparell autònom i intel·ligent; És més, ho considero un plus cap al meu futur professional.

Amb aquesta idea en ment, els projectes oferts eren limitats. La meva idea inicial era la de dissenyar algun sistema de control per a algun prototip de forma que incorporés un microprocessador per a realitzar la tasca. Els primers treballs que se'm varen ocórrer eren sistemes electromecànics amb la necessitat de control com la funció de pèndol invertit en equilibri o una estructura en forma de biga o taula (diferenciades per els eixos de control) capaces de mantenir-se en equilibri amb un pes afegit. L'únic problema era que ambdós projectes ja s'havien realitzat a l'escola i refer-los implicava modificar-ne molts aspectes o millorar-los. No va ser fins que em vaig posar a fer recerca de GEO localització que no em vaig adonar que aquest treball complia amb moltes de les meves expectatives.

Si s'enfocava de la manera que ho vaig fer i que s'explica més endavant, aquest projecte necessitava d'un sistema hardware que havia de ser programat i comunicat a distància amb una tecnologia molt utilitzada. A més a més estava ple de possibilitats per on decantar-se. Va ser en aquest punt que vaig decidir posar-me en contacte amb el meu ponent per a explicar-li la meva idea a veure si ell la veia factible.

1.2. Motivació

Quin grau millor que el meu per a dissenyar un aparell electrònic que al final portés a terme una funció o acció d'ajuda a la humanitat?

Aquest projecte tot i que força extens, té un gran avantatge i és que és molt general, obert i ampliable. Una de les principals motivacions alhora de fer aquest treball és que la GEO localització és

enormement utilitzada. GPS, Smartphones, automòbils, rellotges intel·ligents entre altres dispositius i aparells utilitzen aquesta tecnologia, i les aplicacions i usos que se li poden donar són moltes. Ens trobem doncs amb un tema tot i que ja força explotat, amb molt de futur per endavant.

Un altre dels punts que em va agradar més era que era un projecte funcional, a més no s'ha fet res semblant en tota la carrera, i ho considero molt interessant. D'aquest factor també s'extrau que si en un moment determinat del projecte era impossible continuar ja per la complexitat, pel pressupost o se'm feia preferible continuar-lo per un altre camí, sempre podria redirigir-lo al ser tant obert. La possibilitat d'afegir-hi un servidor, sensors i actuadors, una aplicació mòbil... Està ple de possibilitats per a enfocar i/o ampliar.

1.3. Requeriments previs

Abans de registrar el TFG cal assegurar-se que es podria fer. Això no vol dir no trobar-se amb cap complicació ni problema, ni tant sols detallar-ne totes les parts i buscar-ne possibles dificultats. Però hi ha una tasca que podia resultar impossible de resoldre i caldria buscar la manera de com fer-la si es vol començar amb aquest treball, el receptor GPS i la manera de com transmetre informació.

Dissenyar un receptor GPS és, a part que extens com per un treball de fi de grau, difícil de fer i trobar-ne informació. La poca informació que s'ha trobat era insuficient i molts fòrums recomanaven comprar-ne un ja fet.

La segona tasca era trobar la manera més adequada per a transmetre la informació. La gran majoria de dispositius similars que es troben en venda al mercat funcionen amb una transmissió de dades via Bluetooth. El problema d'aquesta transmissió és la distància a la que és capaç de comunicar-se. Bluetooth és una connexió directa entre un emissor i un receptor que han d'estar suficientment a prop per a poder transmetre's dades de forma eficient. Actualment es poden trobar mòduls Bluetooth 4.0 amb una distància de fins a 100 metres entre emissor i receptor. La tecnologia utilitzada devia de ser una altra.

La solució s'ha trobat pensant amb quins dispositius són capaços de rebre informació es trobin on es trobin, els dispositius mòbils per exemple. Fent una mica de recerca s'ha trobat uns mòduls que utilitzen una targeta SIM que són capaços de comunicar-se amb una empresa de telecomunicacions per a transmetre i rebre dades.

Amb aquests dos blocs tancats, es veu més clara la idea del dispositiu i quins elements contindria i ja es pot posar un nom al projecte per a poder ser registrat i descrit com a un dispositiu hardware que és capaç de rebre senyals GPS per al seu posicionament sense importar on es trobés aquest, ja que

utilitzaria aquesta tecnologia GPRS per a comunicar-se amb un servidor o amb un dispositiu mòbil directament, això ja es discutirà més endavant.

2. Introducció

Al llarg d'aquest document, es presenta la redacció de la memòria d'un Treball Final de Grau dut a terme per a finalitzar els estudis superiors d'Enginyeria Electrònica Industrial i Automàtica; Grau cursat a la Escola Universitària d'Enginyeria Tècnica Industrial de Barcelona EUETIB, tot i que aquest últim any aquesta escola ha quedat integrada a la nova Escola d'Enginyeria de Barcelona Est EEBE.

El treball final de grau plantejat en aquesta memòria tracta sobre la GEO localització. Es planteja, dissenya i crea un dispositiu capaç de localitzar-se i transmetre'n la informació a un servidor per a ser captat per una aplicació mòbil sense importar a quina distància es trobin cap dels elements involucrats.

Per a aconseguir aquest objectiu final, es dividirà el projecte en un seguit de parts ben diferenciades entre elles. En aquest capítol es descriuran. També es troba l'abast del treball.

2.1. Objectius del treball

El treball final de grau ve marcat per tres objectius principals:

Primer de tot cal que el disseny final funcioni. En aquest cas concret, el dispositiu haurà de transmetre a un receptor final les dades del posicionament concret en el instant desitjat de l'aparell i ser mostrades en una interfície d'usuari en un mapa.

Un altre dels objectius importants d'aquesta memòria és que qualsevol persona amb un cert grau de coneixements sobre electrònica pugui ser capaç de desenvolupar el mateix projecte seguint aquesta mateixa com a pauta. Pel que tota la informació ha de ser descrita de manera clara i entenedora però sense esser redundant o hi hagi sobre informació.

Per últim, el software que controla tot el procés ha de ser robust, sense errors i averies cada poc temps. La idea és que el prototip creat pugui fins i tot interessar a possibles empreses que volguessin produir un model en massa basat en el que es dissenyarà en el treball.

2.2. Abast del treball

És important deixar ben marcat l'abast del treball abans de començar a desenvolupar-lo com si d'un treball oficial demanat per a una empresa es tractés. Tot i que no anirà signat ni escrit amb un document a part, si que se'n mantindrà la idea.

- Realització d'un dispositiu capaç de rebre dades de la seva localització mitjançant coordenades GPS.
- Realització d'un dispositiu capaç de transmetre dades a llarg abast.
- Tractament de les dades mitjançant un microprocessador.
- Aconseguir rebre les dades del dispositiu hardware en un servidor o en un dispositiu mòbil.
- En cas d'utilitzar un servidor, programar-lo correctament per a enllaçar les dues plataformes.
- Creació d'una aplicació mòbil simple capaç d'adquirir la informació rebuda.
- Ampliar l'aplicació per a aconseguir que l'usuari no tingui la necessitat d'entendre el funcionament d'aquesta i la manera com es transmeten i reben les dades i que aquestes es mostrin en un mapa per a poder ser interpretades de manera eficient.

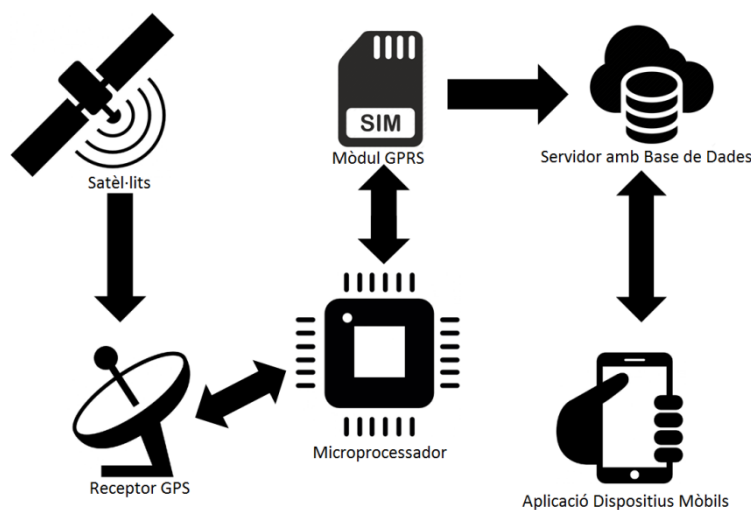


Figura 2.1. Gràfic amb l'objecte del projecte.

Per a fer més visual el funcionament del projecte s'ha dissenyat aquest gràfic amb tots els elements que contempla l'aplicació.

Cal remarcar que aquests abast és el marcat inicialment. Si el projecte s'acaba abans de l'entrega, es començaran un seguit d'ampliacions que farien un projecte més elaborat, però el temps és molt limitat i aquest abast ja té moltes coses a tocar. Es valorarà la feina feta a les conclusions.

Com a possibles ampliacions:

- Afegir detalls i una estructura agradable a l'aplicació juntament amb altres funcionalitats com la pròpia localització del mòbil.
- Afegir al dispositiu hardware creat sensors com un giroscopi i un acceleròmetre per a aconseguir dades reals del comportament del dispositiu.
- Tractar i transmetre la informació dels sensors a la aplicació mòbil juntament amb les dades de localització.
- Fer que l'aplicació sigui capaç de mostrar més d'una posició, incloent la possibilitat de mostrar-ne una demanant una hora en concret.
- Possibilitat de simular una zona on podria dirigir-se el dispositiu en funció de les últimes dades rebudes.
- Afegir un micròfon i altaveu per a poder trucar al dispositiu.

3. Fonaments

És important entendre alguns dels conceptes i la història de molts dels elements que després seran necessaris per a la creació i desenvolupament del projecte. Aquest apartat es centra amb aquests temes que, si ve no són imprescindibles per a fer el treball, seran d'utilitat alhora de comprendre tot el que el rodeja. Ve a ser, amb poques paraules, la teoria del treball abans de fer-ne la pràctica.

3.1. Global Positioning System

El Sistema de posicionament global o *GPS*, es tracta d'un sistema de navegació global per satèl·lit que proporciona informació sobre GEO localització i de temps a qualsevol receptor GPS sempre i quan hi hagi una línia de visió entre el receptor i quatre o més satèl·lits GPS. És comú relacionar el sistema GPS amb la trigonometria, i és perquè aquesta tecnologia funciona amb les lleis de la trilateració.

3.1.1. Característiques tècniques del sistema

El sistema actual està compost per 24 satèl·lits que orbiten en 6 òrbites diferents. Es troben a una altitud de 20200 km de la superfície terrestre i tarden 12 h en donar una volta completa a la Terra. Tots aquests satèl·lits es troben inclinats 55 graus respecte l'equador terrestre i presenten una vida útil d'uns 7.5 anys.

La freqüència de la senyal portadora és variable si és d'ús civil (1575.42 MHz) o militar (1227.60 MHz). La precisió que ofereixen és oficialment de ± 15 m de marge d'error en el 95% del temps, tot i que generalment aquest marge es troba entre els ± 3 m.

Com que aquests satèl·lits incorporen rellotges atòmics, també poden donar la hora i de manera molt precisa, amb resolució de nano segons. També cal detallar que la capacitat d'usuaris utilitzant aquesta tecnologia és il·limitada [1].

3.1.2. Altres sistemes de navegació: GLONASS, Beidou i Galileo

Si no són suficients 24 satèl·lits de altíssim valor econòmic, tant Rússia, la República Xinesa com Europa volen fer la seva pròpia constel·lació de satèl·lits; De fet Rússia ja presenta la seva pròpia constel·lació activa actualment.

En aquest treball però s'utilitzaran exclusivament els satèl·lits de EEUU tot i que hi ha maneres d'utilitzar també altres constel·lacions amb els receptors i guanyar així amb precisió.

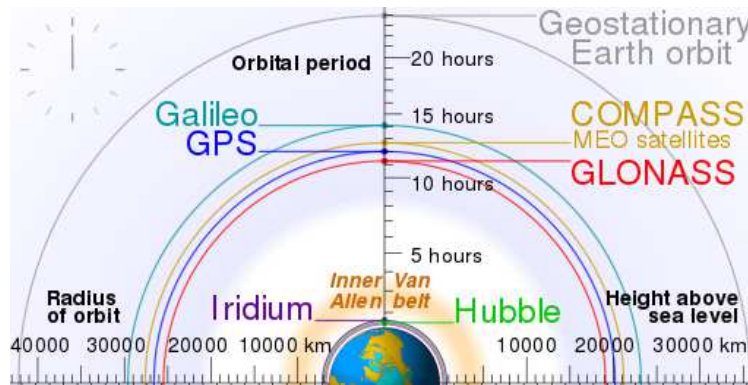


Figura 3.1. Figura amb la posició de diferents constel·lacions de satèl·lits. (Font: Satellite Navigation [2]).

3.1.3. Funcionament del sistema i del receptor GPS

La manera com els satèl·lits es comuniquen amb els receptors o, més ben dit, com els receptors reben i interpreten les dades enviades pels satèl·lits és força interessant.

Pel que fa al satèl·lit, cada transmissió dura 30 segons i porta 1500 bits de dades codificades. Aquesta petita quantitat de dades està codificada amb una seqüència pseudoaleatòria (PRN) d'alta velocitat que és diferent per a cada satèl·lit. Els receptors GPS coneixen els codis PRN de cadascun i per això no només poden descodificar el senyal sinó que la poden distingir entre diferents satèl·lits.

Les transmissions són cronometrades per començar de forma precisa en el minut i en el mig minut tal com indiqui el rellotge atòmic del satèl·lit. La primera part del senyal GPS indica al receptor la relació entre el rellotge del satèl·lit i l'hora GPS. La següent sèrie de dades proporciona al receptor informació d'òrbita precisa del satèl·lit.

La informació que realment és utilitzada pels receptors GPS s'anomena efemèrides, que es tracta de dades del satèl·lit com la seva posició a l'espai, la seva hora atòmica, informació doppler...

Com ja s'ha exposat abans, el receptor ha de utilitzar la trilateració per a determinar la seva posició. De manera anàloga a la triangulació, la trilateració juga amb el coneixement de dos o més punts de referència i la distància entre aquests i el subjecte analitzat. Per a determinar la localització relativa d'un subjecte en un pla tridimensional, es necessiten al menys quatre punts de referència, en el nostre cas, quatre satèl·lits. Resulta confós que hi hagi bibliografia que cregui que són tres els satèl·lits suficients per a donar les coordenades d'un objecte, però no és així [3].

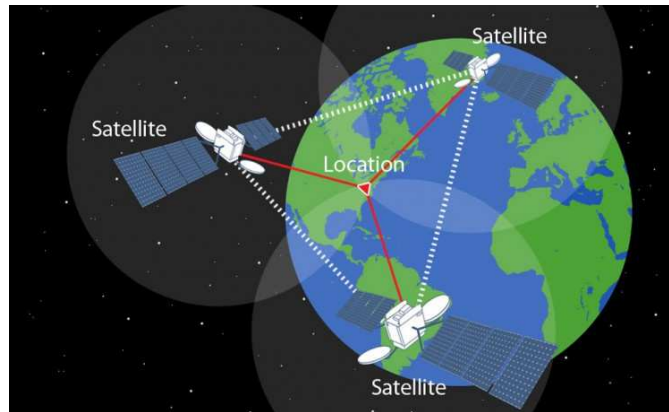


Figura 3.2. Figura de localització amb tres satèl·lits. (Font: David Pérez [4])

Amb tres satèl·lits es pot determinar la localització d'un receptor en un pla bidimensional. I són quatre els necessaris per a donar les coordenades en un pla tridimensional. Les matemàtiques i la física aplicada en tot el sistema, són impactants.

Tots els satèl·lits incorporen un rellotge atòmic imprescindible per a la seva localització. Tot i que no s'entrarà amb molts detalls, aquest rellotge actua amb les lleis de la física quàntica, amb els nivells energètics dels electrons i l'energia que s'allibera en el canvi d'aquests nivells. Aquest canvi és instantani i sempre de la mateixa energia basada amb la relació de Plank.

$$\Delta E = h \times f \quad (\text{Eq. 3.1})$$

L'increment d'energia d'un electró ve donada per la constant de Plank i per la freqüència que ve determinada per el tipus de material utilitzat per a fer el rellotge atòmic. El Cesi per exemple oscil·la a una freqüència de 9.192631770 GHz, pel que cada segon un rellotge atòmic de Cesi variarà electrons d'una capa o nivell energètic a un altre 9 bilions de vegades. El que el fa inigualable amb precisió però, és que qualsevol àtom de Cesi de tot l'univers oscil·la a la mateixa freqüència. [5]

La pregunta recau en perquè es necessita una precisió tant acurada del temps si el que es vol saber es la posició d'un objecte a la Terra?

Sabent que les ones de ràdio emeses pel satèl·lit viatgen a la velocitat de la llum, si un receptor és capaç de saber en suma precisió en quin moment han estat enviades les ones, podrà saber on es troba el satèl·lit amb la següent senzilla fórmula:

$$d = c \times \Delta t \quad (\text{Eq. 3.2})$$

On d és la distància del satèl·lit, c és la velocitat de la llum i l'increment de temps que ha viatjat la ona. Amb una mala precisió del temps, multiplicada per una constant molt elevada i fent el mateix procés amb tres satèl·lits més, el problema de precisió seria molt elevat.

És més, també s'ha de tenir en compte un altre condicionant: La teoria de la Relativitat. No entrarem en molts detalls, animo a qualsevol lector que en faci una petita cerca per internet de la teoria.

Com que un observador en terra veu els satèl·lits en moviment, el seu rellotge atòmic es veurà afectat per aquesta teoria. Exactament, seran uns 7 microsegons per dia els que es rebaixaran a causa de l'efecte de la dilatació del temps del seu moviment relatiu. Però és que encara hi ha més! Els satèl·lits estan en òrbites altes per sobre de la Terra, on la curvatura de l'espai-temps a causa de la massa de la Terra és menor del que està a la superfície d'aquesta. Un altre dels postulats de la teoria de la relativitat esmenta que el temps passa més lent a prop d'un objecte massiu que els situats més lluny. Per tant, els rellotges dels satèl·lits estaran marcant més ràpid que els rellotges idèntics sobre la Terra. Un càlcul utilitzant la relativitat general prediu que els rellotges de cada satèl·lit GPS han d'anar per davant dels rellotges terrestres 45 microsegons per dia.

La combinació d'aquests dos efectes relativístics significa que els rellotges a bord de cada satèl·lit haurien de marcar uns 38 microsegons per dia extres ($45 - 7 = 38 \mu s$). No sembla gaire, però el sistema GPS funciona amb una precisió de nanosegons, i 38 microsegons són 38.000 nanosegons. Si aquests efectes no es tenen en compte adequadament, els errors en les posicions globals continuarien acumulant-se a una velocitat d'uns 10 quilòmetres cada dia! [6]

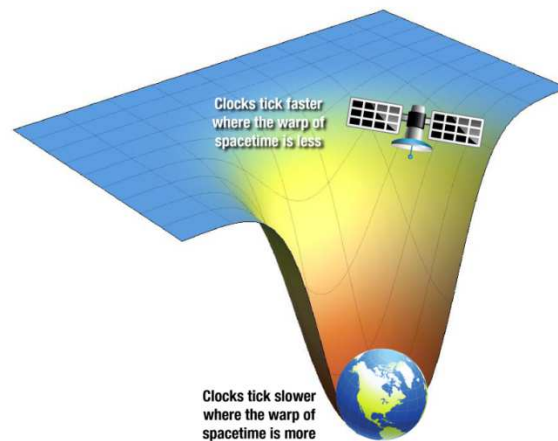


Figura 3.3. Figura sobre la relativitat afectada pels satèl·lits. (Font: Write Science [7])

És important saber que de models de receptors n'hi ha molts, alguns millor que d'altres. Un dels elements diferenciadors entre ells és el nombre de canals. Nombre de canals és el nom que se li ha

donat a nombre de correladors, la funció d'aquests és la de desmodular una freqüència codificada en particular. Per a entendre-ho però cal una explicació més profunda.

Un receptor GPS rep l'espectre de totes les freqüències enviades pels satèl·lits a la vegada. Per a trobar i aïllar una freqüència en concret, aplica el mètode de correlació; Genera una seqüència específica d'un dels satèl·lits i la correlaciona amb una senyal d'entrada (input signal) i si coincideixen, el receptor ha trobat un satèl·lit. Aquest procés però pot arribar a fer-se molt llarg ja que si no coincideix amb la senyal rebuda, ha de canviar la seva senyal per un bit i tornar-ho a intentar fins a haver passat pels 1023 bits i no hagi trobat un satèl·lit. En aquestes circumstàncies, canvia la senyal del satèl·lit en un període diferent i ho trona a provar.

Per si no hi havia suficient, també s'ha de tenir en compte l'efecte doppler de les ones emeses pel satèl·lit. Aquestes poden arribar a variar fins a ± 10 kHz i això fa que s'hagi de buscar variant-ne la freqüència abans que el correlador es doni per vençut i passi a buscar a una altra senyal de satèl·lit.

Si el correlador és capaç de buscar en 1000 transmissions per segon, serà capaç de analitzar un codi de 1023bits per segon. Com hi ha 32 PRN diferents, el màxim temps que podria estar-se un correlador buscant una transmissió seria de uns 32 segons sense contar l'efecte doppler, el qual podria augmentar el temps significament [8].

3.1.4. L'antena

L'antena és un component que porten incorporat els receptors GPS per a poder captar més eficaçment les senyals emeses pels satèl·lits, recordem que aquests es troben ni més ni menys que a una altitud de 20200 km.

Actualment les antenes són fabricades de ceràmica majoritàriament, i presenten forces avenços generació a generació de tecnologies d'aquest component. Ara ja són capaces de rebre senyals dins dels edificis, tot i que en els nuclis de ciutats com Nova York on hi ha molts rasca cels, la tasca és més difícil. Sempre són més eficaces si s'enfoquen cap a cel obert.

Citant paraules de U-blox: Un receptor GPS només assolirà el rendiment desitjat si la relació senyal/soroll (C / N_0) dels satèl·lits és d'almenys 44 dBHz. En un sistema ben dissenyat, aquesta relació es troba de mitjana entre 44 dBHz i 50 dBHz, i amb una antena activa, s'aconsegueixen fàcilment 47 dBHz [9].

Requeriments d'una bona antena:

- Baix nivell de directivitat
- Bona visibilitat del cel



- Alt guany
- Un bon filtre

La directivitat d'una antena és una mesura que indica com de direccional és aquesta. Direm que una antena té un baix nivell de directivitat (proper a 1) quan és capaç de irradiar de igual manera en totes direccions. La fórmula que descriu la directivitat relaciona la densitat de potència màxima en una direcció amb el valor mitja de potència de la superfície d'aquesta. El valor d'aquesta relació pren valors de 1 a infinit i s'expressa normalment amb dB al ser adimensional.

El guany d'una antena té dos conceptes, un quan es transmet i un altre quan es rep informació. Com que en el projecte no es transmetrà, es pot definir com a guany com de eficaçment es converteix les ones de ràdio rebudes en energia elèctrica.

El guany relatiu, es defineix com la relació entre la densitat de potència radiada a una direcció i la densitat de potència que radiaria una antena isotròpica, en igualtat de distàncies i potències lliurades a l'antena. Per a explicar-ho amb unes altres paraules, si s'agafa una antena que irradia igual en totes direccions i es compara la potència radiada amb la antena objecte d'estudi en igualtat de condicions, en trobarem el guany.

En aquest projecte es rebran senyals de ràdio en direccions aleatòries i diferents pel que, tot i que hi ha antenes amb una directivitat molt tancada, per exemple en aplicacions on l'emissor i el receptor siguin fixes, aquest no és el cas i s'hauria de buscar una antena que no fos direccional.

El problema que l'antena no sigui direccional, és que rebrà més interferències al captar qualsevol ona electromagnètica. D'aquí ve a que es necessiti un bon filtre. La funció del filtre és el de filtrar totes les freqüències i deixar passar només les establertes pels components del filtre. D'aquesta manera només es podran rebre les senyals dels satèl·lits teòricament. El filtre desitjat serà un filtre passa banda o band-pass-filter a una freqüència L1 de 1575.42 Hz. També es sol veure filtres SAW, especials per a ones electromagnètiques que incorporen un cristall piezoelèctric per a convertir primer la ona en una mecànica per després ser reconvertida en elèctrica amb elèctrodes.

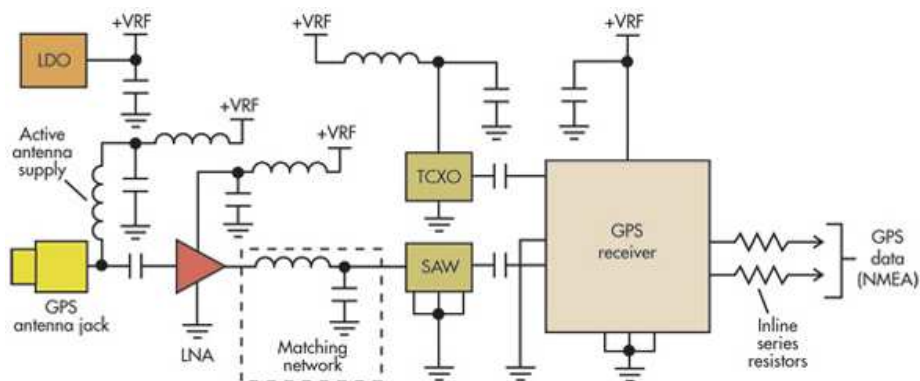


Figura 3.4. Proposta d'esquema elèctric d'una antena activa juntament amb el receptor GPS i altres components importants com l'oscil·lador i el filtre SAW. (Font: Electronic design [10])

En aquest exemple hi podem observar l'antena, una etapa amplificadora, un filtre SAW i el receptor GPS. El bloc amb referència TCXO és un oscil·lador compensat per temperatura.

Existeixen dos grans tipus d'antena; Antenes actives i antenes passives.

Les antenes passives són aquelles que només contenen l'element radiant, és a dir l'estructura ceràmica o en forma d'hèlix. Són aquelles que no estan alimentades i per tant no tenen guany. Les antenes actives presenten un amplificador de baix soroll i això les beneficia respecte de les altres a canvi que s'han d'alimentar. Molts receptors GPS només admeten antenes actives.

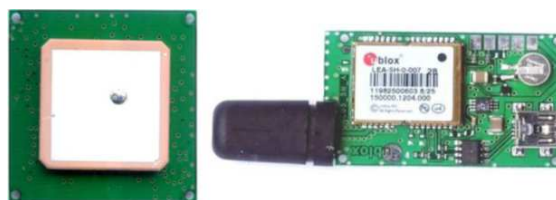


Figura 3.5. Formats Patch i Hèlix d'antenes passives . (Font: U Blox [9])

Les antenes passives són les més utilitzades perquè les aplicacions on es necessiten poques dimensions, eficàcia moderada i baix cost i consum són les més abundants [9].

3.2. Transmissió de dades

Cal detallar primer de tot que el que es pretén buscar en aquest apartat no són les maneres estàndard de transmissió de dades en la indústria, ni les interfícies per a fer-ho, etc.

Ha de complir amb les condicions que l'emissor es pot arribar a trobar a l'altre punta de la superfície terrestre del receptor i tot i així ha d'aconseguir rebre les dades desitjades sense múltiples errors. Per a complir-ho, es passarà directament a investigar sobre transmissió sense fils.

3.2.1. Xarxes sense fils de poc abast

Si ens centrem directament amb les xarxes sense fils, trobem que aquestes es poden dividir amb diferents subcategories que es classifiquen amb funció de l'abast a la qual poden comunicar-se.

En un principi es va optar per a fer el treball amb una xarxa del tipus Bluetooth, altament utilitzat en aplicacions com per exemple de domòtica. La companyia de Bluetooth llançarà al mercat en els propers mesos la nova tecnologia Bluetooth 5.0 que ja està sent afegida a dispositius mòbils que sortiran al mercat el proper any 2017. Tot i que aquesta cinquena generació resulta molt interessant

(i de fet inaccessible en el moment de desenvolupament del treball), no és factible per a ser utilitzada en aquest projecte. L'abast màxim d'aquesta tecnologia de comunicacions, tot i quadruplicar-se respecte la seva anterior generació, no supera els 100 metres amb repetidors. Això és perquè Bluetooth és una xarxa Wireless Personal Area Network o *WPAN* que s'utilitzen per a comunicar dispositius de uns 10 metres de distància [11].

Les xarxes *WLAN* com per exemple la ret Wi-Fi presenten un abast màxim de fins a 20 km i utilitzen ones de ràdio per a transmetre la informació.

Tot i així, encara queden dues tecnologies sense fils que no s'han mencionat amb més abast que les anteriors, anomenades *WMAN* i *WWAN*.

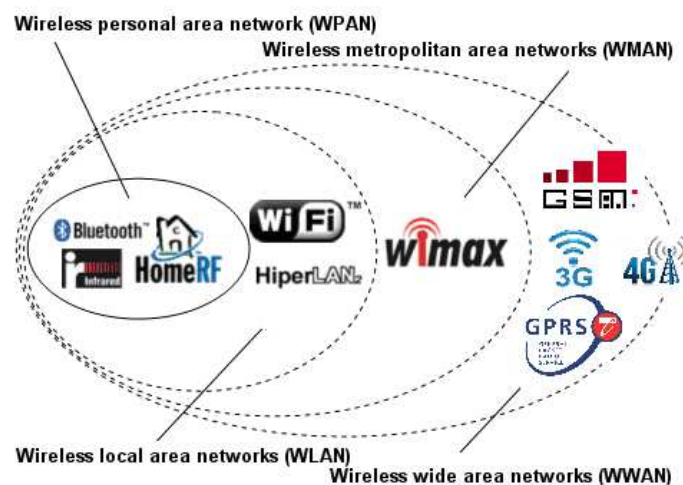


Figura 3.6. Esquemàtic amb les diferents tecnologies i marques de xarxes sense fils.

3.2.2. Xarxes sense fils de llarg abast

Les sigles *WMAN* són utilitzades per a referir-se a Wireless Metropolitan Area Network que significa Xarxa sense fils d'àrea metropolitana. Es tracta de xarxes d'alta velocitat que donen cobertura a un àrea geogràfica força extensa però limitada. Poden abastar ciutats i es poden connectar diversos nodes formant xarxes entre sí.

Podem trobar dins d'aquest sector tecnologies basades amb WiMax, que funciona per microones de freqüències compreses entre 2.5 i 5.8 GHz. Aquest tipus de xarxes s'utilitzen sobretot en campus, per a interconnectar blocs de pisos i edificis d'una mateixa ciutat o en zones on el desplegament de cable resulta de un cost massa elevat per la densitat de població que hi viu (zones rurals per exemple).

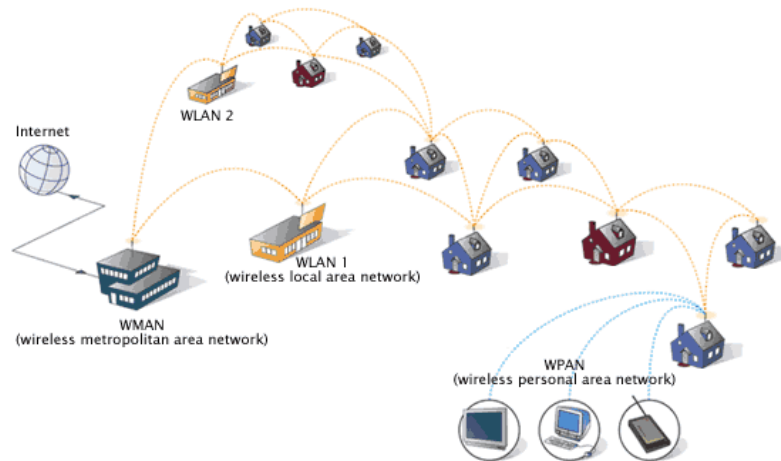


Figura 3.7. Esquemàtic on es contemplen les tres xarxes descrites en conjunt. (Font: Redes Inalambricas TESCO ISC [12])

L'última xarxa sense fils i la més àmplia s'anomena *WWAN*, *Wireless Wide Area Network* que significa Xarxa sense fils d'àrea ampla. Utilitza tecnologies com GSM, GPRS, UMTS, 3G i 4G o EDGE i tracta una xarxa informàtica formada per conjunts de xarxes més petites com WLAN moltes de les quals són instal·lades per els proveïdors d'internet per proveir de connexió als seus clients.

Aquest tipus de xarxes s'utilitzen per a transmetre dades a través de llargues distàncies i entre diferents xarxes i arquitectures més petites. Al igual que la WLAN, la WWAN utilitza ones de ràdio per a transmetre la informació a llargues distàncies. Un exemple podria ser el tenir una WWAN que connecta a diferents xarxes LAN d'empreses de tot el món.

També utilitzen sistemes com el de distribució local multipunt (LMDS) o Wi-Fi per a proveir la xarxa de connexió a internet. En el següent apartat s'estudiarà la transmissió de les ones, com són capaces d'abastir tanta distància i seguir essent fiables i les tècniques que s'utilitzen per aconseguir-ho [13].

3.2.3. Modulació de les ones de ràdio

Com ja s'ha explicat en els apartats anteriors, tant les xarxes WLAN com WWAN utilitzen les ones de ràdio per a transmetre la informació desitjada. Però la eficàcia de transmetre informació mitjançant aquestes ones és limitada i molt inferior a la comunicació per cable però compta amb la avantatge de no necessitar un medi físic per a propagar la senyal i això les fa increïblement útils. El funcionament de la transmissió d'informació és interessant.

El procés s'anomena modulació d'ones i es tracta de modificar o modular les ones portadores de tal manera que se'n variï una característica d'aquestes i en funció del valor, se'n pugui extraure informació. Cal entendre que la ona portadora serà la base a la qual es superposarà certa modulació

que variarà en funció de la senyal que es vulgui transmetre, la qual s'anomena ona moduladora. La imatge 3.8 al final del apartat pot ajudar a entendre el concepte.

Les ones portadores, que no són més que ones electromagnètiques, es poden modular analògica i digitalment en funció del paràmetre que es vol transmetre. A més, hi ha diferents tècniques de modulació que són les següents.

3.2.3.1. Modulació analògica.

- Modulació per amplitud (AM). L'amplitud (intensitat del senyal) de l'ona portadora és variada en proporció a la forma d'ona que es transmet. Aquesta forma d'ona correspon, per exemple, a sons que seran reproduïts, o la intensitat de llum dels píxels d'una televisió. La avantatge de aquesta modulació és que la seva desmodulació (s'ha de desmodular la senyal portadora quan arriba al receptor per a que la informació superposada pugui ser interpretada) és senzilla i això permet que els receptors siguin econòmics
- Modulació per freqüència (FM). En aquest cas la característica de la ona que és variada és la freqüència. La diferència entre la freqüència modulada i la nominal de la portadora és proporcional amb la senyal que es vol transmetre. Les aplicacions més usals on s'aplica aquest tipus de modulació són per a transmetre missatges de veu, so i àudio. La ràdio en seria un bon exemple.
- Modulació per fase (PM). En aquesta modulació es varia la fase. De manera molt similar a la modulació per freqüència, (de fet és impossible distingir-les sense saber la senyal moduladora) es tracta d'una modulació exponencial. La diferència recau en els punts on hi ha la màxima desviació de freqüència. Mentre que a la FM es troba quan l'amplitud de la senyal portadora i moduladora disten el màxim (és a dir quan una es troba en el punt més positiu i l'altre en el més negatiu), en la PM la màxima desviació es dona quan la senyal moduladora passa per zero, és a dir, la desviació de la freqüència és proporcional a la derivada de la senyal modulant [14].
Aquest tipus de modulació és menys utilitzada que les anteriors ja que requereix de sistemes i equips més complexos però presenta algunes avantatges com una reducció dels soroll i un ús més eficient de l'energia.
- Altres mètodes de modulació. Existeixen altres mètodes de modulació que estan basats en els tres anteriors. Aquests són alguns dels més destacables:
 - o Modulació d'amplitud en quadratura (QAM), que correspon a modular dues ones portadores en amplitud de igual freqüència però desfasades 90° . Finalment les dues ones són sumades i dona com a resultat una modulació en amplitud juntament amb una de fase amb el resultat de poder transportar dos senyals independents amb la mateixa ona.

- Modulació per una sola banda lateral. Aquesta modulació és una millora de la AM. Amb un equip receptor adequat, és possible disminuir l'energia emesa pel emissor reduint l'ample de banda de l'ona modulada. L'equip desmodulador haurà de ser capaç de extraure la senyal amb la meitat de la ona, que podrà ser la part alta o la baixa. [15]

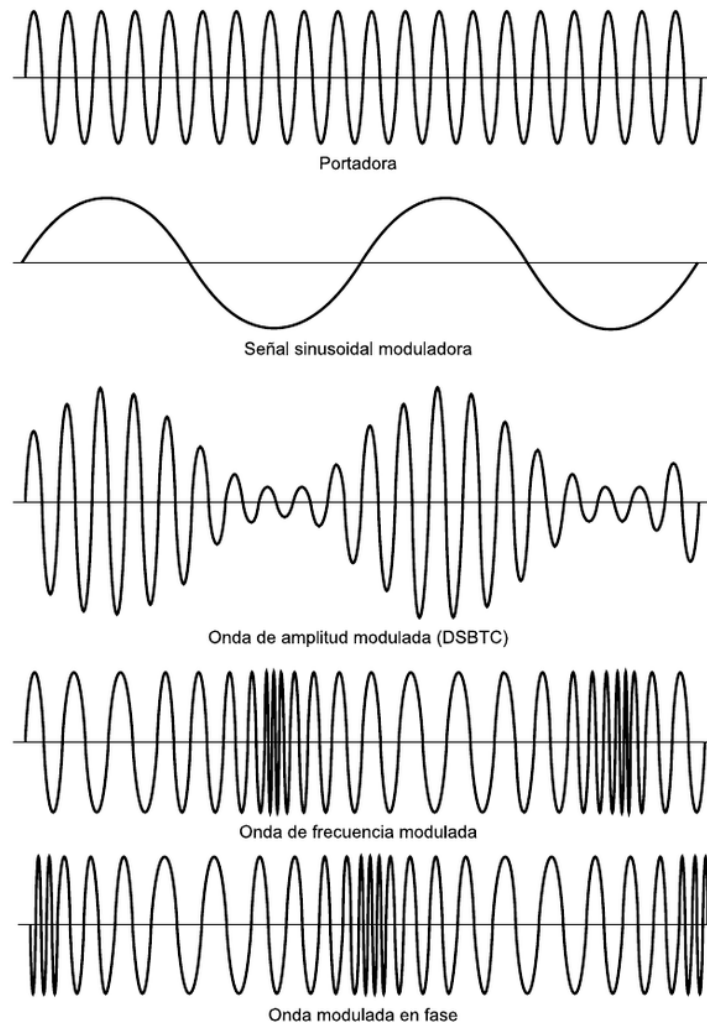


Figura 3.8. Esquema amb les diferents modulacions analògiques. (Font: WIKI BOOKS [16])

3.2.3.2. Modulació digital

- Modulació per desplaçament d'amplitud (ASK).

Com a la modulació analògica, la digital també es pot variar per amplitud. En aquest cas el nombre de valors que es sobreposaran a la senyal portadora seran dos. Una amplitud més

gran que una altra expressaran els valors binaris. El desmodulador buscarà constantment una certa amplitud que li donarà valor 1, si no es compleix li donarà valor 0.

- Modulació per desplaçament de freqüència (FSK).

Dos freqüències donen valors discrets a aquesta modulació.

- Modulació per desplaçament de fase (PSK).

En aquesta modulació angular, es fa variar la fase de la portadora amb valors discrets. Cada cop que hi ha un canvi de valor discret, es canvia la fase de la senyal modulada tal i com s'observa a la imatge 3.9.

- Altres mètodes de modulació. Hi ha molts altres mètode de modulació digital tots basats en els tres primers descrits però el més important és el següent:

Manipulació On-Off (OOK). Una millora de la ASK és la digitalització total de la senyal modulada. Amb aquesta tècnica el nombre 1 és representat per una certa amplitud i el valor 0 és representat per una amplitud nul·la. D'aquí el terme On-Off. Cal dir que la major part de modulació per amplitud es fa amb aquesta tècnica ja que permet un estalvi d'energia considerable i no representa gaire més dificultats ni cost el seu tractament [13].

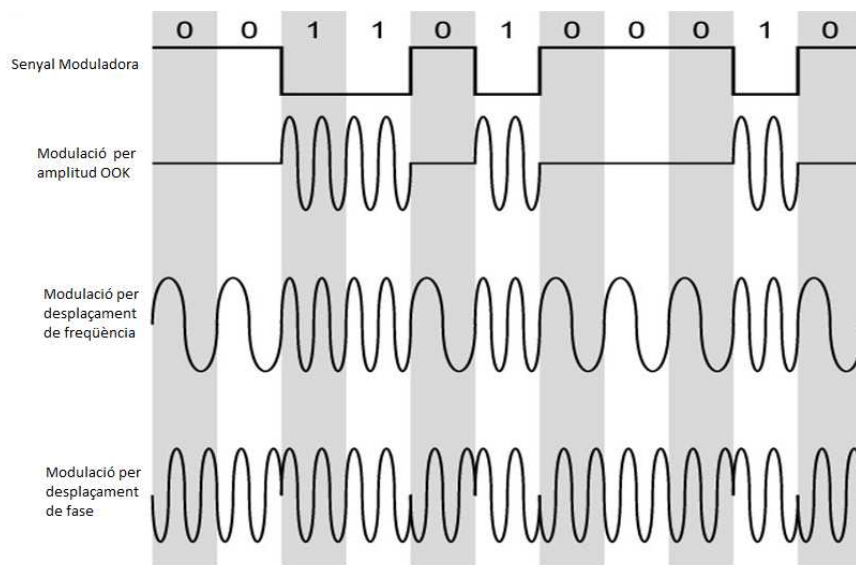


Figura 3.9. Esquema amb diferents modulacions digitals. . (Font: WIKI BOOKS [16])

3.2.4. Desmodulació

La desmodulació d'una ona inclou totes les tècniques utilitzades en telecomunicacions per a recuperar la informació modulada prèviament amb una ona portadora per un emissor. Com hi ha diferents tècniques de modulació d'una ona, també hi ha d'haver de diferents per desmodular-la.

Un dels mètodes més pràctics per a desmodular una ona modulada en amplitud és utilitzant circuits de detector d'envolupant, que consisteix amb l'adquisició de la superfície de la ona entrant mitjançant un circuit RC que pot ser molt simple.

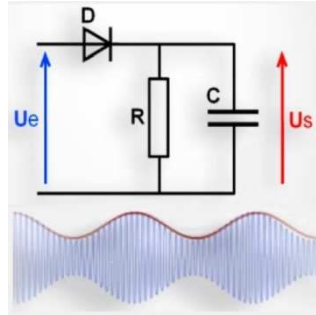


Figura 3.10. Circuit i senyal d'entrada (en blau) i sortida (en vermell) d'un desmodulador d'amplitud analògic. (Font: edumedia [17])

El circuit complet seria més extens però tampoc és d'interès.

La modulació per freqüència necessita de circuits més complexes per a la seva desmodulació, però té la avantatge que presenta millor immunitat del soroll i una major fidelitat.

Hi ha diverses maneres de fer-ho, les més habituals, el detector de pendent, el discriminador Foster Seeley, el detector de relació, desmodulador de PLL i el detector en quadratura. Els tres primers tracten discriminadors de freqüència de circuit sintonitzat. La funció d'aquests és convertir una senyal modulada FM en AM, és a dir converteixen en variació de tensions les variacions de freqüència que entren pel receptor. Així després ja només fa falta afegir-hi un desmodulador de AM per aconseguir la senyal original.

La modulació de fase és una variant de la modulació en freqüència pel que les tècniques de desmodulació són els mateixos que els anteriors recalcant els desmoduladors PLL o de fase tancada. Aquest desmodulador no requereix de circuits sintonitzats i automàticament compensa els canvis en la freqüència de la portadora degut a la estabilitat en el oscil·lador de transmissió [18].

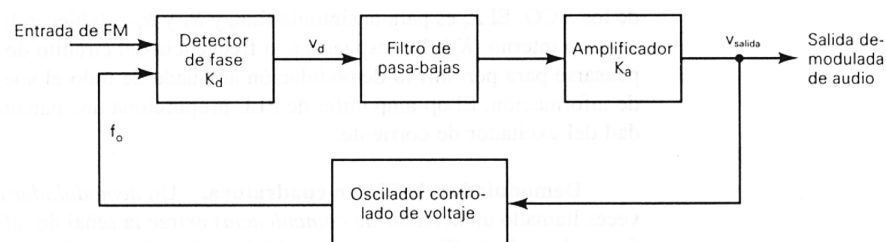


Figura 3.11. Diagrama de blocs per a un desmodulador d'angle amb PLL. (Font: profesores.edu [18])

3.2.5. Tecnologies o generacions de les xarxes de comunicacions mòbils.

Les xarxes sense fils WWAN són les que més abast contemplen. S'ha vist que funcionen per mitjà d'ones de ràdio de baixa freqüència però hi ha hagut diverses generacions que han portat a aquestes xarxes al que són avui en dia, una tecnologia insubstituïble.

Primera generació o 1G.

La més antiga de totes, també coneguda com TACS a Espanya i engloba les tecnologies mòbils analògiques utilitzades fins als anys 80. El sistema de comunicació TACS permetia la transmissió de només veu (incapaç de transmetre dades). A partir d'aquí varen sorgir un conjunt de protocols en diferents països amb les seves incompatibilitats.

Segona generació o 2G.

Es crea el sistema GSM (Global System for Mobile Communications) capaç de transmetre veu però que també ja permet la transmissió de dades a baixa velocitat, a 9.6kbit/s. Aquest estàndard surt al mercat l'any 1990 amb el nom de GSM-900. Al 1992 ja hi havia els primers telèfons mòbils amb aquesta tecnologia que permetia a part de la comunicació, enviar missatges de text curts (SMS). El sistema Europeu GSM opera en torn als 900 MHz i defineix un sistema complet que inclou no tant sols la interfície ràdio, sinó també una completa arquitectura de xarxa, el que permet el desenvolupament de multitud de nous serveis sobre l'estàndard GSM. Tot i així, aquest estàndard estava pensat fonamentalment per a la transmissió de veu pel que presentava certes limitacions com una lenta connexió, poc ample de banda i un cost elevat en la transmissió de dades.

Extensió de la segona generació o 2.5G.

El pas entre la segona i la tercera generació de tecnologies de comunicacions mòbils va ser lenta. En aquest període entre els 2000 i mitjans del 2003, es varen llançar al mercat una de les ampliacions més reconeguda de GSM, el que es coneix per a General Packet Radio Service o GPRS. El servei es cobra per el volum de data transferit contrastant amb la commutació de circuits on es paga per el temps que el canal de comunicacions està actiu.

Aquest nou servei oferia velocitats de dades de 56 kbit/s fins a 114 kbit/s i a part de poder utilitzar-se per enviar missatges curts SMS, també oferia d'enviar missatgeria multimèdia o MMS i la possibilitat de comunicació per internet com el email i l'accés a la web.

Per altra banda també es van treure al mercat el HSCSD, un servei de commutació de circuits de GSM homologada per l'institut d'estàndards de telecomunicacions europeu, i la tecnologia EDGE, un altre desenvolupament de GSM que permet velocitats de fins a 384 kbit/s. Aquesta última es podria

considerar la precursora del UMTS aparegut en la tercera generació de xarxes de comunicacions mòbils.

Tercera generació o 3G.

El pas a la tercera generació va ser lent, de fet, al 2010 encara es podien trobar dispositius utilitzant la tecnologia de segona generació o 2.5G. Tot i això, ens remuntem a mitjans del 2003, moment que es llança la primera xarxa de tercera generació per a ús comercial. El nom que més destaca en aquesta nova generació és el de UMTS (Universal Mobile Telecommunications System) que és una tecnologia que a part de la millora substancial amb velocitat de transferència de dades multimèdia i connectivitat a la xarxa, seguretat i transmissió de veu de millor qualitat, sinó que també permet el roaming internacional i la interoperabilitat. El que vol dir que un terminal amb tecnologia de tercera generació es pot connectar a qualsevol xarxa 3G de tot el món. Estem parlant que UMTS és capaç d'arribar als 2 Mb/s en condicions ideals de funcionament, unes cinc vegades més ràpid que EDGE de 2.5G que per cert, encara és molt utilitzat quan un terminal no troba cap enllaç a una xarxa 3G. És fàcil detectar-ho perquè el dispositiu mostra a la part de connectivitat una lletra E en comptes de 3G o 4G [19].

El 3G ultra ràpid o 3.5G.

Quan a la pantalla del nostre terminal apareix una H o una H+ significa que estem connectats a una xarxa HSDPA (High Speed Downlink Packet Access). A l'any 2006 la velocitat de transferència de dades se'ns havia quedat curta i surt al mercat aquest nou servei que ofereix velocitats de baixada de fins a 14 Mb/s i de pujada de 5.8 Mb/s. I una nova millora d'aquest servei s'anuncia a l'any 2008 que permetia un augment de fins a 84 Mb/s de pujada i 22 Mb/s de baixada en condicions ideals. La velocitat de transferència de imatges, vídeos amb alta qualitat, música, era increïblement alta.

La nova generació, el 4G.

L'any 2008 la ITU (International Telecommunications Union) defineix els requisits dels estàndards de comunicació 4G i el que més es fa notar és el de les velocitats de baixada i de pujada, que han de ser d'almenys de 100 Mb/s en moviment i de 1 Gb/s en estat estacionari. Totes les tecnologies que complissin amb aquests requeriments de velocitat amb una alta qualitat i seguretat, podrien publicitar-se com a xarxes 4G. La quarta generació està basada en protocols d'internet i només s'aconsegueix amb la convergència de xarxes de cable i de sense fils. Hi ha també algun altre aspecte com el de ser capaç de compartir i utilitzar els recursos de la xarxa per donar suport a més usuaris simultanis forma dinàmica o el traspàs de xarxes heterogènies de manera suau (amb el 3G es nota el canvi de xarxes quan, per exemple, ens movem amb transport. Hi ha un temps que és impossible accedir a la xarxa per aquest motiu).

Un dels aspectes més importants de la tecnologia 4G és l'eliminació de nodes de la xarxa de commutació de circuits i de commutació de paquets paral·lels utilitzant el Protocol d'Internet versió 6 (IPv6). L'estàndard usat actualment, IPv4, té una limitació finita en el nombre d'adreces IP que es poden assignar als dispositius. IPv6 proporciona un nombre molt més gran d'adreces disponibles, i serà fonamental per a proporcionar una experiència optimitzada per als usuaris.

El 4G també utilitza tècniques complexes de multiplexació i de transmissió de les ones com són "Múltiple entrada, Múltiple sortida (MIMO)" o "multiplexació per divisió de freqüències ortogonals (OFDM)". L'arquitectura de les xarxes 4G també es veu millorada.

Un dels noms que més destaquen amb la quarta generació és les sigles LTE Advanced, corresponents a Long Term Evolution Advanced. LTE és un projecte d'estàndard de 3GPP (grup d'associacions de telecomunicacions de tercera i ara quarta generació) que volia diferenciar-se de les xarxes 3G ja àmpliament extenses al 2010 que fos capaç de superar-les i seguir tenint la avantatja competitiva sobre les tecnologies mòbils. Avui en dia aquest estàndard és el més utilitzat per a la adquisició de 4G per les tècniques ja explicades abans i el compliment de les especificacions. El seu gran rival és WIMAX2 tot i que aquest no arriba a tanta velocitat i no és compatible amb tecnologies 3G si fos el cas que són les úniques disponibles mentre que LTE sí. LTE Advanced per resumir és doncs, el que UMTS era per 3G però per a 4G.

Aquesta tecnologia tant puntera té una basant fosca però relacionada amb la comercialització i el màrqueting que han volgut exprimir les empreses de telecomunicacions. La tecnologia no era capaç d'arribar als mínims de velocitat encara però pel fet d'utilitzar bandes de freqüència diferent de la tecnologia 3G, algunes empreses com Vodafone van anunciar el 4G per a forçar el canvi de molts terminals, publicitar aquestes tecnologies i guanyar molts més beneficis. Molts parlen del 3.9G d'aquest procés. Però és que encara hi ha més perquè actualment, al 2016 s'han aconseguit arribar als 300 Mbps en algunes zones de més alta densitat de xarxes LTE i les companyies les anomenen ja el 4G+.

Sense cap dubte encara queda camí per a explotar aquesta tecnologia i per a poder parlar de futures generacions, tot i que ja se sap que aquesta avança molt ràpid i les companyies de telecomunicacions de ben segur que faran tot el possible per a ser les pioneres en aquest camp que mou xifres desorbitades i és el futur de la comunicació [21], [23].

La generació del futur o 5G.

Sense saber encara si hi haurà una generació entremig que enllaçarà la recent quarta generació amb la cinquena, es preveu que una nova generació es comenci a desenvolupar a partir del 2020. Si més

no, la 4G es va anunciar el 2008 i totes les generacions passades es porten entre sí una dècada de interval. No és estrany per això, poder trobar informació de 5G.

Es preveu (perquè encara no hi ha definits uns estàndards per aquesta futura generació) que les velocitats en les que s'enfocarà el 5G siguin de entre 1Gbps a 5 Gbps. També aspira a permetre un major nombre d'usuaris de banda ampla connectats a la xarxa per unitat de superfície, i que el consum de dades pugui ser molt més alt o il·limitat.

Sigui com sigui, de ben segur que el nom de 5G arribarà al mercat més aviat del que tocaria per a poder treure'n benefici igual com ha passat amb el 4G.

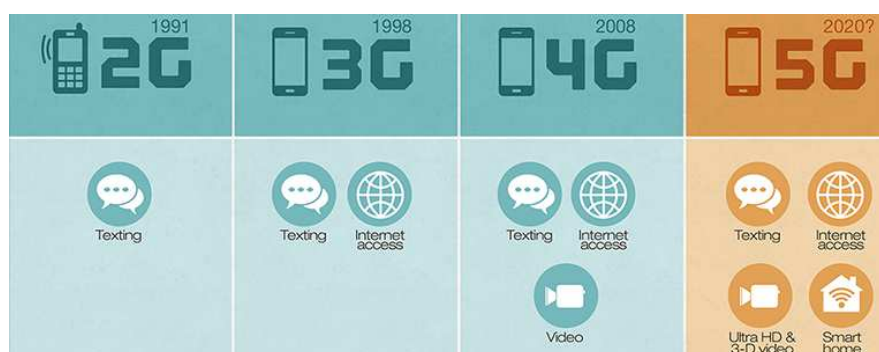


Figura 3.12. Esquema de les diferents generacions i el seu abast. (Font: Expansion [23])

3.3. El Microprocessador

Si bé tenim receptors GPS que són capaços de rebre les senyals dels satèl·lits del espai i xarxes de telecomunicacions que poden transmetre informació arreu del món amb un dispositiu adequat, aquests i una infinitat més de transductors, actuadors, etc. No tindrien importància sense l'element que els interpreta i els hi dona les ordres de funcionament, el microprocessador. Però, què és exactament aquest component?

El microprocessador és el component més important en qualsevol equip de control automàtic, gestiona totes les dades del sistema i realitza les operacions que li pertoca en funció de com estigui programat. Es tracta doncs del element central del equip molts cops expressat com el cervell del sistema.

Aquest IC (circuit integrat) està format per milions de transistors increïblement petits que són els encarregats de donar la informació binària amb la que treballen tots els equips electrònics digitals. Això s'aconsegueix fàcilment controlant-los per la seva base en mode de commutació, és a dir fent-

los treballar en tall i en saturació només, i evitant la zona lineal que a més és la que més potència consumeix.

També és important saber que el microprocessador funciona amb operacions senzilles, lògica binària, operacions aritmètiques simples i té accés una memòria volàtil anomenada principal on hi guardarà els valors de totes les operacions que faci i en llegirà les instruccions guardades. A partir d'aquestes operacions i amb la seva combinació, s'aconsegueix la programació a la qual ens trobem acostumats a treballar.

En el principi dels microprocessadors, els programadors d'aquests dispositius havien de dissenyar els programes amb codi binari, per sort després van anar apareixent diferents llenguatges que utilitzaven codi per a programar-los el qual després un compilador passava tot el codi programat a codi binari per a què el microprocessador pogués entendre-ho. Fa uns anys el llenguatge més habitual era l'ensamblador, un llenguatge de baix nivell amb el qual es podien fer instruccions bàsiques com sumes amb un acumulador, increments, salts en el registre entre d'altres. Amb tots ells però es podia dissenyar qualsevol codi que es pot dissenyar avui en dia amb un llenguatge de més alt nivell, però amb moltíssimes més línies de codi.

Tot i així encara hi ha empreses que utilitzen aquest codi per a programar els seus microprocessadors. La raó per la qual no s'han passat al llenguatge de més alt nivell és perquè quan més alt és el nivell del codi, més instruccions farà el propi compilador per arribar al mateix punt ja que un programador pot decidir com escriure el codi per a què ocupi el mínim de línies possibles, i això té un gran avantatge, la velocitat de processament. En feines on es requereix un control molt ràpid, per exemple en control de curtcircuits en motors trifàsics, quan més ràpid funcioni el microprocessador, millor.

A excepció del comentat anteriorment, la majoria de programadors de microprocessadors utilitzen un llenguatge de més alt nivell com pot ser C i les seves variants. Tampoc s'entrarà en detalls, però Arduino utilitza una variant del C++, un llenguatge que és el predecessor del C.

Arduino és una marca registrada. En aquest treball apareixerà sovint aquest terme, els crèdits del nom i la placa van dirigits a aquesta marca. Per a més detalls, visitar les seves polítiques a la web oficial.

3.4. Emmagatzematge de dades

Ja es pot intuir que farà falta una recollida d'informació a una memòria per a la realització del projecte. El pas es troba just un cop s'han aconseguit les coordenades, aquestes s'hauran

d'emmagatzemar per a després poder tractar-les. En aquest apartat es veurà totes les opcions i què farà falta per a que funcioni.

Podem dividir aquest apartat amb quatre categories diferents en funció del lloc i del tipus de dades:

- Emmagatzematge local. Aquest tipus tracta de guardar tota la informació que es va rebent a una memòria no volàtil allotjada amb el propi sistema, és a dir, poder afegir una memòria externa com podria ser una micro SD a Arduino per a que aquest enregistres totes les dades en aquesta i després ja tractar-les i enviar-les.
- Emmagatzematge a un servidor o en el núvol. Aquesta opció consta de l'enviament de les dades a un servidor extern connectat a internet per a ser guardat en aquest. En altres paraules, les dades serien enviades a un ordinador (exactament a un espai de memòria d'un disc dur o SSD d'una companyia que ofereix aquest servei) on quedarà guardat allà fins que s'elimini.
- Emmagatzematge en un fitxer. La manera la qual es pot emmagatzemar la informació ja sigui local o en el núvol, pot ser del tipus fitxer. Aquest és el típic procediment per a guardar imatges, vídeos, documents de text, text pla (.txt) entre d'altres.
- Emmagatzematge en una base de dades (BBDD). S'entén per base de dades a un espai amb un conjunt de dades d'un mateix context organitzades, estructurades i enllaçades per a la seva interpretació posterior. Normalment només enregistren dades alfanumèriques.

Si ens fixem amb els avantatges, inconvenients i necessitats de cadascun podem trobar el tipus que millor s'adapta a les nostres necessitats.

Quan es parla d'emmagatzematge local en vers de al núvol, la diferència més important és que mentre un tens la informació disponible en tot moment ja que aquesta es troba guardada a una memòria a la qual el microprocessador en té un accés directe, a l'altra necessites una connexió en línia per a poder agafar-la i enviar-la. Per altra banda, és també més fàcil i ràpid emmagatzemar les dades en una memòria local, però amb el que de veritat treu avantatja l'enregistrament en un servidor en línia, és que les dades es mantindran allà sense importar que el hardware del aparell sofreixi alguna anomalia com la corrupció de dades de la memòria. A més a més, si es guarda la informació en un servidor, qualsevol altre dispositiu podria obtenir-la donant així dos besants cap a aquest tema; per una banda algú amb connexió a internet podria rebre la informació enviada pel dispositiu, i per altra, que qualsevol informació enviada a través d'internet i penjada en un servidor pot ser robada per una tercera persona.

Per altra banda, si ens fixem en la manera d'emmagatzemar la informació, en un fitxer o en una base de dades, la principal diferència recau en la manera en com es troba aquesta informació disponible per a ser tractada. Mentre en un fitxer, la informació es troba poc ordenada i gens estructurada, en una base de dades és tot el contrari. Si l'interès es troba en emmagatzemar la informació estructuradament aquesta n'és la millor opció. El problema està en si es vol emmagatzemar informació que no sigui del tipus text, amb la qual cosa s'haurà d'utilitzar un fitxer per a fer-ho.

Per acabar, cal dir que a la realitat els problemes i dispositius actuals ofereixen en molts casos situacions híbrides on coexisteixen memòries locals i en línia en funció de la informació que es vulgui guardar i de l'aplicació en concret, i d'igual manera amb les BBDD i els fitxers. Per a posar un exemple, la coneguda aplicació de mòbils Whatsapp n'és un bon comprovant; La llista de contactes, grups d'amics i missatges els té guardats en una base de dades mentre que les imatges, vídeos, GIFs els envia com a fitxers, i a més a més, tota la informació de l'aplicació la pots visualitzar sense connexió a internet ja que es guarda a la memòria interna del telèfon, però també s'envia a un servidor de la companyia (parlem de naus industrials plenes de ordinadors amb discs durs on s'enregistra la informació de tots els usuaris durant un cert temps) per a que des d'allà la informació pugui ser transmesa als altres usuaris de l'app i, en cas de pèrdua o canvi de mòbil, poder recuperar tota la informació, converses que tenies en un cert temps.

Per acabar comentar que l'enregistrament híbrid és més complexa de dur a terme ja que inclou ambdós sistemes però dona una versatilitat que no pots aconseguir d'una altra manera.

Com a conclusió doncs sobre les necessitats d'emmagatzematge del projecte, aquest haurà de tenir informació ben estructurada i només seran caràcters, i a més haurà de ser capaç d'enviar-li la informació a un dispositiu mòbil pel que aquest també haurà de rebre les dades i un servidor en línia per a enllaçar les dos plataformes seria la opció més evident a priori. Per tant doncs ens centrarem amb la creació d'un servidor i la seva pujada i baixada de dades, i amb les bases de dades per a guardar la informació en el servidor ordenadament per a facilitar la feina a l'aplicació del mòbil.

3.4.1. El servidor

Un servidor web no és res més que un ordinador connectat a internet capaç de rebre i transmetre informació a la xarxa 24h al dia, 365 dies a l'any. Per a fer més precisa aquesta definició, s'ha de pensar que no es tracta d'ordinadors comuns, aquests estan dissenyats específicament per a emmagatzemar informació i transmetre-la tant eficientment com sigui possible. Empreses gegants com Google amb les quals entren filials com l'enorme plataforma d'allotjament de vídeos Youtube (comprada per l'empresa l'any 2006 per 1650 milions de dòlars), necessiten una impensable quantitat de memòria i servidors per satisfer les necessitats actuals mundials. Per a posar un exemple, cada minut es pugen a Youtube unes 400h de gravació, i això fan més de 550.000 hores al

dia. Fent un senzill càlcul, una persona tardaria a veure tot el contingut penjat a la plataforma en un dia ni més ni menys que 64 anys. (Dades extretes d'una pàgina web oficial de Google [24])



Figura 3.13. Un dels magatzems de centre de dades de Google. (Font: Google, Inc. [25])

Per tant doncs, la funció dels servidors és la de emmagatzemar fitxers i emetre'ls per internet per a poder ser visitats pels usuaris. Quan un usuari entra en una pàgina web, el que realment està passant és que el seu navegador es comunica amb el servidor demanant-li que li ensenyi certes dades que té allotjades aquest. Aquest procediment es dona gràcies a que tots els ordinadors tenen un nombre IP únic i cada cop que es connecten dos per a transmetre's informació, es demanen entre ells aquest nombre de tal manera que si hi ha 1000 usuaris demanant informació a un mateix servidor, aquest sàpiga a quin d'ells va cada una amb la direcció IP de cadascun.

S'ha parlat d'empreses multinacionals, però també trobem l'altre besant, no totes les pàgines d'internet necessiten de tants servidors ni memòria per a operar, tot depèn de les seves necessitats, i també hi ha la possibilitat de llogar una part d'un servidor comú si la teva pàgina web no necessita moltes prestacions. Bé, realment pots llogar un servidor més potent i gran per a pàgines més grans pagant una suma més alta a la companyia distribuïdora de servidors.

3.4.2. Família de protocols d'internet

Per a que internet funcioni és necessari un conjunt de protocols que estipulen la comunicació entre els ordinadors de la xarxa. Sense aquests protocols no seria possible l'enviament i recepció de dades correctament, la codificació d'aquestes juntament amb la direcció IP, entre moltes altres funcions. De vegades se li denomina conjunt de protocols TCP / IP, en referència als dos protocols més importants que la componen, que van ser dels primers a definir-se, i que són els dos més utilitzats de la família.

Abans d'explicar alguns dels protocols més importants d'internet, cal entendre el funcionament d'aquesta família. Primer de tot cal saber que aquest estàndard es divideix en quatre capes, i en cadascuna d'elles (de més alt a baix nivell) hi actuen diferents protocols per a desenvolupar el conjunt d'operacions necessàries per a la utilització de la xarxa. Tenim la capa d'aplicació, on es defineixen les aplicacions de xarxa i els serveis estàndard d'internet a l'abast de l'usuari. Aquesta capa utilitza protocols com HTTP o FTP i és la de més alt nivell. Seguidament, s'utilitza la capa de transport, amb protocols com TCP per a enviar i rebre dades. No confondre amb la capa d'internet, la qual busca la ruta per a transportar els paquets creats per la capa de transport per la xarxa entre routers i utilitza protocols com IP. Finalment la capa física és la que s'encarrega de transmetre els paquets de forma local, del emissor i receptor al router. Aquesta última capa utilitza protocols com Ethernet per a direccionar la informació.

Es podria fer una analogia d'enviament de paquets entre dos ciutats. L'emissor porta el paquet fins a l'oficina de transport local de la ciutat a la capa física, allà la capa de transport l'empaqueta i li posa un nombre de direccionament o port a utilitzar per a poder identificar de quina manera s'ha de tractar el paquet, diguem per exemple si és una carta, un paquet fràgil, un de molt gran, etc. Tenint en compte que l'enviament es fa en qüestió de segons, la oficina haurà d'esperar una resposta del receptor i també ha d'enviar un paquet indicant que ja no hi ha més paquets a enviar per a assegurar-se que tot ha arribat on ho havia de fer i, novament, esperar resposta. La oficina doncs, passa els paquets a l'aeroport que li afegirà la direcció del receptor dins de tota la xarxa d'aeroports mundial i s'encarregarà de determinar la ruta més òptima per a fer arribar els paquets en cada moment. Aquest nou nivell o capa d'internet no s'assegura que els paquets arribin, però si arriben, el receptor ja n'enviarà uns altres al emissor, que també portarà per la ruta més òptima aquesta capa i la de transport sabrà si han arribat correctament o no. Finalment la oficina de transport de la ciutat del receptor portarà els paquets al receptor altre cop per la capa física. Falta veure la capa de més alt nivell, la d'aplicació que, realment aquesta dicta com es troben els paquets. Si utilitzem el port 80 voldrà dir que els paquets són de hipertext i si utilitzem el 20 o el 21 estarem enviant arxius en text pla, és a dir sense xifrat. Podríem dir que és el llenguatge en el que es troba la informació que després la capa de transport dividirà en paquets [26].

Nota del redactor: He decidit incloure una analogia perquè opino que aquest temari és molt dubtós, la informació que es troba per la xarxa poc entenedora i molts cops es contradiu. Una analogia com aquesta pot facilitar la comprensió del funcionament de la xarxa per a qui llegeixi aquest treball.

Per a poder realitzar la comunicació entre el dispositiu i el servidor i després del servidor a l'aplicació mòbil, s'haurà d'utilitzar tot aquest conjunt de protocols que passarem a explicar amb més detall a continuació.

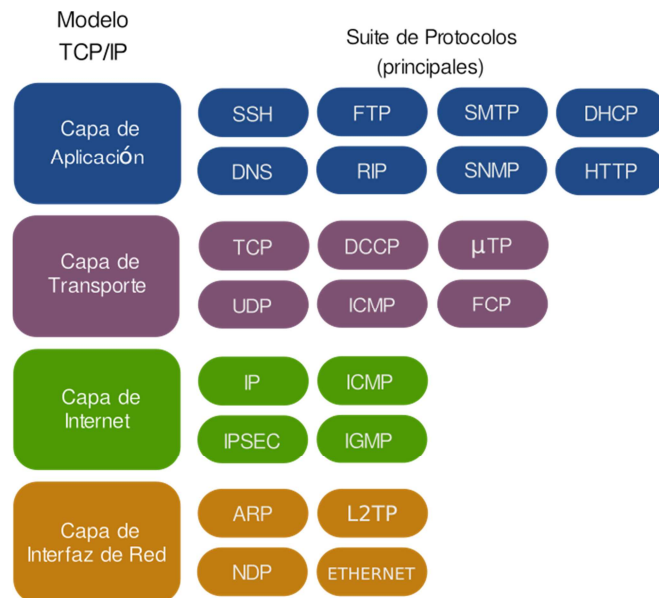


Figura 3.14. Capes del model d'internet i els protocols més importants. (Font: Wikipedia[27])

3.4.2.1. Transmisión Control Protocol

El protocol de control de transmissió o comunament anomenat TCP, és l'encarregat de comunicació entre ordinadors. La principal funció del protocol és que aquesta operació es realitzi de forma segura, sense pèrdues i lliure d'errors. La manera com opera el protocol és:

Primerament estableix la comunicació entre els dos ordinadors, amb el que s'anomena *three way handshake* amb el qual primer es determina si el port de comunicacions del servidor en el que el client intenta realitzar la comunicació es troba obert, en tal cas retornarà una resposta afirmativa, sinó la resposta contindrà el bit de RST en 1 i es tornarà a provar la connexió en un altre port. A més a més, a cada pas de connexió s'envia una seqüència determinada per a que no es pugui falsejar la comunicació.

Un cop comprovada la comunicació, comença l'enviament de dades entre client i servidor de manera ordenada. Per al enviament de dades utilitzant aquest protocol s'utilitzen bits, bytes i seqüències de control com el checksum, bit de paritat, CRC, per a evitar seqüències errònies, duplicades, etc.

Finalment es tanca el port de comunicacions i la connexió entre els dispositius amb quatre passos similars als d'inici. Es pot donar el cas que un hagi acabat la connexió i l'altre encara estigui enviant dades. Per això es necessiten quatre passos un per a demanar la fi de la comunicació i un per a acceptar la finalització de l'altre terminal.

El protocol TCP inclou, com ja s'ha dit la seguretat de la connexió en el transcurs de la transferència d'informació, a més, s'han anat fent variacions al llarg dels anys del protocol adaptant-lo a les noves necessitats i problemes fins a convertir-lo en un dels protocols més importants i robusts de comunicacions.

En conclusió doncs, aquest protocol s'encarrega de dividir els fitxers en paquets de dades, amb la robustesa i seguretat de la comunicació entre els dos nodes, i finalment reconstrueix el fitxer en el lloc de destí per a poder rebre la informació correctament [27], [28], [29], [30].

3.4.2.2. Internet Protocol

El protocol d'internet també conegut com a IP està molt relacionat amb el protocol anterior TCP però aquest està més enfocat a la capa de la xarxa i no pas en la de transport de dades.

L'objectiu del protocol és l'entrega dels paquets de dades de la millor manera possible, afegint la direcció IP als paquets per al enrutament de dades i realitza un control d'errors de la seqüència.

És important saber que aquest protocol no s'encarrega de saber si la transmissió de dades ha resultat o no exitosa; Per a aquesta funció ja hi ha el protocol TCP.

La diferència entre el protocol TCP i el IP doncs, tracta que el primer divideix el fitxer a enviar en paquets, s'encarrega de que la transmissió amb el receptor sigui correcte, i el segon afegeix als paquets de dades una direcció de destinació i emissió (direccions IP) i a més a més fa recorren la xarxa per la ruta més eficient fins al node receptor, que pot ser diferent amb cada paquet però sempre intentarà garantir la més eficient entenent per eficient el camí més curt o menys congestionat per tràfic de dades [27], [29].

3.4.2.3. Hipertext Transfer Protocol

Comunament anomenat com a HTTP, aquest protocol és el que permet la transferència d'arxius a través de la xarxa (text, imatges, vídeos, arxius multimèdia...). Utilitza el port 80.

En aquest projecte ens basarem en els arxius de text ja que són els que s'ha de tractar. HTTP defineix una sèrie de mètodes per a fer peticions els quals utilitzarem en absolutament les tres interfícies que comunicarem a través d'internet. Les més importants es mostren a continuació:

- GET: Quan un client fa una petició GET a un servidor web, si aquest és públic o el client està autoritzat, el servidor li retorna totes les dades per les quals ha estat programat per enviar quan se li faci aquesta petició en aquesta direcció.

GET és més interessant del que sembla perquè també permet enviar paràmetres al servidor, per exemple coordenades, afegint al final de la URL els paràmetres amb certes condicions.

- POST: Just al contrari del GET, POST s'utilitza per a penjar alguna cosa al servidor. És la pròpia programació del servidor que definirà què fer quan li arribi aquest recurs. Actualitzar o crear objectes és la causa més freqüent de POST.
- DELETE: Acompanyat normalment del nom d'un recurs del servidor, aquesta comanda elimina algun objecte del servidor, per exemple un usuari, una aplicació, un missatge, etc.
- PUT: Similar a POST, PUT no s'utilitza per a crear cap recurs, sinó per a actualitzar-ne un de ja creat o per a pujar-ne de nous. Per a una millor comprensió, utilitzaríem POST per a crear un nou usuari en una pàgina web, i PUT per a pujar fotografies.

Tot i que hi ha més peticions i el protocol HTTP s'ha anat millorant en versions, aquestes cinc són les més utilitzades.

Existeix també un protocol idèntic a HTTP però més segur anomenat HTTPS on la S afegeix la paraula Secure. Està destinat, com el seu nom indica, a realitzar connexions d'enviament de dades, cargues, creacions, etc. De manera segura encriptant les dades entre l'emissor o el client i el receptor o el servidor web. Aquest protocol és molt utilitzat actualment per pàgines importants de la xarxa i per a direccions que demanin dades bancaries o nombres de targetes de crèdit per a qüestions òbvies. Estrictament parlant, HTTPS fa referència al protocol anterior però li afegeix una capa de connexió segura (SSL) [31].

3.5. Introducció a programació de mòbils

Comencem ja un bloc nou en el projecte, el de la programació de l'aplicació que farà d'intermediari entre el projecte i l'usuari. Aquest tema és molt complex i cal entendre molt bé, hi ha molts detalls a tenir en compte i moltes coses que s'aniran aprenent al llarg de la creació de l'aplicació.

3.5.1. Les plataformes mòbils

En aquest apartat s'explicarà com funciona, en general, el món de la creació d'aplicacions mòbils juntament amb les possibles plataformes a les quals es poden penjar aquestes. Recalcar que actualment les que predominen en el mercat són Android, una plataforma que l'any 2005 va ser comprada per Google i basa el nucli del seu sistema operatiu amb Linux, i iOS, una plataforma exclusiva de la famosa marca Apple amb un sistema operatiu de nom macOS que deriva de Unix. Va rebre la seva darrera actualització, la 10.12 el juny de 2016. Doncs aquest mercat és un duopoli que lideren aquestes dues marques, i tot sembla que seguirà essent així durant un bon temps.

3.5.1.1. Especificacions tècniques d'android i les diferents versions

Android és un sistema operatiu de la marca Google basat en Linux, però les especificacions d'aquesta plataforma no s'acaben amb una frase i resulta interessant, pel fet d'utilitzar-la per a la nostre aplicació, aprendre sobre el seu funcionament, la història que l'ha portat fins a dalt de tot i què li espera en el futur.

Tot i que l'empresa fundadora d'aquesta marca va ser Android inc., la marca no va començar a donar els seus fruits fins que va ser comprada pel gegant d'internet l'any 2005. No va ser però fins al 2008 que es podria comprar un telèfon intel·ligent amb aquest software, però el creixement des d'aleshores és sorprenent.

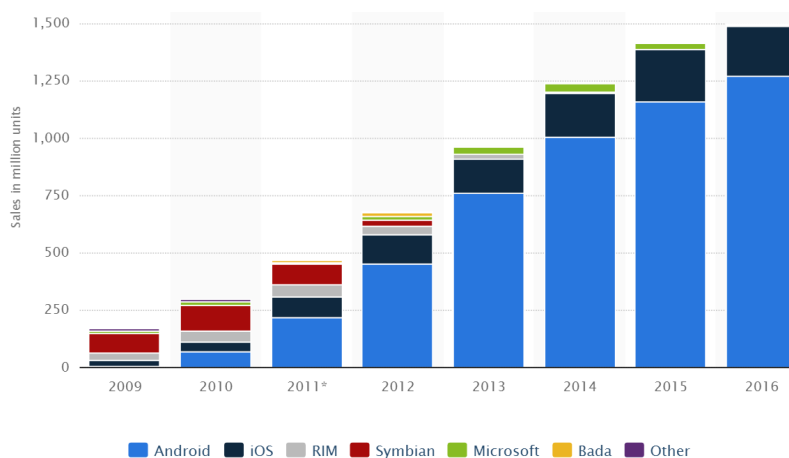


Figura 3.15. Gràfica comparativa de mòbils i el seu sistema operatiu.. (Font: Statista [32])

En la gràfica exposada, es pot veure el nombre d'unitats de Smartphones enviats globalment en milions els últims anys. Com es pot observar també figuren les plataformes a les quals han estat fabricats en els que es remarca Android en primera posició i IOS en segona. També es pot notar que el mercat al 2016 era només cosa de dos i la resta de plataformes són, gairebé inexistents confirmant així el duopoli dels dos gegants del mòbil.

Android té una plataforma de desenvolupament de codi obert, amb el qual tot el seu software pot ser estudiat i modificat per els usuaris. Es troba escrit en tres llenguatges de programació diferents: el nucli en C, algunes biblioteques de tercers, en C++ i per últim la seva interfície d'usuari (UI) està escrita en Java. El nucli d'Android és Linux i és monolític, això vol dir que l'arquitectura del seu sistema operatiu treballa en la seva totalitat en espai del nucli i ell es troba en mode supervisor exclusivament. Aquests sistemes tenen un nucli gran i complex, que engloba tots els serveis del sistema. Està programat de forma no modular, i té un rendiment més gran que un micronucli. No

obstant, qualsevol canvi a realitzar en qualsevol servei requereix la recompilació del nucli i el reinici del sistema per aplicar els nous canvis.

El tipus de mercat el qual va dirigit Android no és exclusiu de Smartphones, tot i que sí que n'és el principal sector, també fa de un sistema operatiu per a tablets, televisions, interfícies de vehicles i rellotges digitals intel·ligents. La interfície gràfica actual és una marca registrada de Google anomenada Material Design i va ser introduïda a la versió 5.0 com es veurà a continuació en substitució de Holo, l'anterior interfície.

Android soporta diferents plataformes com són ARM, una arquitectura RISC (Reduced Instruction Set Computer) que disposa del conjunt d'instruccions de 32 i 64 bits més utilitzat globalment, x86, un conjunt d'instruccions de Intel de la família 8086 entre d'altres. Això ho podrem observar més endavant en la configuració del programari per a la creació de la app.

El sistema de gestió de paquets d'android, ve donat amb l'extensió .APK. Aquestes sigles signifiquen Android Application Package utilitzat per a distribuir i empaquetar components, aplicacions, jocs... per a Android. Realment un arxiu .apk no és més que un conjunt de fitxers necessaris per a Android comprimits en una sola extensió.

En tot aquest temps el sistema operatiu Android ha anat millorant i creixent satisfent les necessitats dels seus usuaris amb diferents versions que anaven treient cada cert temps. La primera versió important de la plataforma va ser Android dònut, el qual afegia el quadre de cerca ràpida per internet, la disponibilitat per a qualsevol mida de pantalla i resolució amb la qual manera es va poder utilitzar aquest sistema operatiu per a dispositius amb una resolució de pantalla diferent a la de 320x480. Per últim, va canviar el disseny d'Android market (l'antic Google play/ Play Store) en el moment en que les aplicacions de tercers de la plataforma començaven a despuntar.

Amb la segona versió d'Android, la 2.1 o Eclair es va focalitzar amb exprimir l'alta densitat de les pantalles per a la creació d'imatges i fons de pantalla interactius de 854x480. A més a més s'afegeix una de les característiques més especials per a tots aquells dispositius amb un receptor GPS integrat, Google Maps Navigation, amb detallades indicacions, vista 3D, indicacions per veu entre d'altres fan canviar la manera com es veien els mòbils intel·ligents del moment. En aquesta versió també es va afegir un sintetitzador de veu al teclat que intentava escriure el que deies pel micròfon del telèfon d'una manera més o menys satisfactòria. Aquesta aplicació es va millorar en la seva següent versió, la 2.2 on se't permetia realitzar les principals funcions del telèfon per veu, com entrar al navegador o afegir una alarma. Va ser també en aquesta versió que es va introduir la funció de zona wi-fi portàtil on el dispositiu es convertia en un Modem per a donar connexió a altres mòbils. Per últim tenim la versió 2.3 la qual va fer un pas de gegant per al desenvolupament de jocs en Android. Es van poder crear jocs en 3D amb bons gràfics donant accés a al so, controls, gràfics i emmagatzemant del telèfon.

La versió 3.0 d'Android va anar centrada per a les tablets. Amb aquesta versió anomenada Honeycomb, es podrien llegir llibres, veure vídeos, explorar mapes, de forma més senzilla i interactiva amb els dispositius de més polzades. També es va afegir una nova barra dels sistemes per a deixar ja definitivament els mòbils amb botons enrere. Per últim també es va incorporar una barra de ajustos ràpids amb el qual es podia accedir a informació important més ràpid com la hora, encendre i apagar el wi-fi o les dades, entrar al menú d'ajustos, abaixar el volum del mòbil, etc.

Android Ice Cream Sandwich fou la versió 4.0 de la plataforma i va tocar funcions de rendiment del sistema, personalització de la pantalla principal podent afegir carpetes entre d'altres. També va introduir el menú de control de l'ús de dades, on es permetia administrar l'ús de dades del dispositiu en funció de la tarifa contractada a l'operadora de telecomunicacions. Per últim a destacar és la incorporació d'android Beam, una eina per a compartir informació multimèdia entre dos dispositius sense necessitat d'obrir menús o aplicacions ni vincular els telèfons. Només feia falta unir-los i tenir NFC (Near Field Communication). NFC basa el seu funcionament en la inducció magnètica. La comunicació entre dispositius es produeix mitjançant l'intercanvi d'informació que provoca la creació de dos camps magnètics generats per antenes. Podríem dir que és una eina similar a Bluetooth. La millora d'aquesta versió, la 4.1 inclouia a la llista de nous components Google Now, una eina pionera en el seu sector que donava informació i assistència com per exemple quin temps feia o quant de temps tardaries a arribar a cert lloc sense ni tant sols haver sortit de casa. També afegia la opció de fer el dispositiu multiusuari, cada conta amb les seves aplicacions, notes, missatges fons de pantalla etc. Acabem la versió amb la 4.4 o Android KitKat. Amb aquesta s'afegeix el OK Google amb el que mencionant aquesta frase podem iniciar una cerca per veu. També s'afegeix el destacar els contactes amb els que més parles i diverses opcions més per a fer més intel·ligent el telèfon.

Android 5.0 o Lollipop la plataforma renova menús i finestres amb Material Design, a més aquesta versió té com a objectiu centrar-se amb els nous rellotges intel·ligents, televisors i tablets per a tenir-ho tot sincronitzat. Parar una cançó per reprendre-la a la televisió o una cerca per internet des del mòbil al cotxe. Per últim apareix una pantalla de notificacions en la pantalla de bloqueig interactiva.



Figura 3.16. Imatge amb diferents dispositius utilitzant la versió 5.0 d'Android (Font: Android [33])

Android 6.0 està enfocat a la seguretat i a un consum més eficient del dispositiu. Millora de Google Now, la possibilitat de desactivar els permisos de les aplicacions i d'elegir què vols compartir amb cada app i quan. La bateria és ara utilitzada d'una manera més eficient i optimitza el dispositiu amb funcions com Descans i Aplicacions Inactives[33].

L'última versió treta al mercat per la companyia és Android 7.0 o Nougat el qual afegeix característiques com que al aixecar el mòbil, s'encendrà la pantalla o la millora del sistema de notifikacions. Una de les incorporacions més destacables és la multi finestra, amb la qual es podrà utilitzar dues aplicacions en pantalla dividida. També soporta de forma nativa la realitat virtual tant de moda actualment, amb aquesta funció, també s'afegeix Daydream, un nou Play Store però exclusivament dedicat a aplicacions i jocs en realitat virtual.

Només queda parlar dels plans de futur que té Android sobre la taula, Android 8.0 o Android O anunciat fa ben poc, a finals de març de 2017 i que encara es troba en fases de desenvolupament. El que es vol aconseguir amb la futura versió és una millor optimització de la bateria limitant més aplicacions en segon pla, redissenyar alguns menús com el de configuració, els icones de les app tindran moviment al ser amb dos capes diferents, tecnologia PIP o Picture in Picture amb la que es podrà veure un vídeo i utilitzar altres aplicacions a la vegada i millores en les notifikacions podent-les ordenar entre d'altres aspectes que o bé encara no s'han enunciat o que estan dissenyant actualment l'equip d'Android [34].

3.5.2. Requisits previs i material necessari

Si bé es pot pensar que amb la descàrrega d'un programa ja es pot començar a dissenyar una aplicació mòbil, això no és així, fa falta primer de tot pensar en alguns aspectes, i complir certes condicions per a que l'aplicació funcioni i a més pugui ser operativa per a qualsevol usuari d'un Smartphone.

El programa base amb el qual es desenvoluparà l'aplicació es diu Qt Creator. S'ha decidit utilitzar aquest programa perquè és amb el que es dissenyen a l'assignatura a la qual s'ha assistit per a la seva creació. El programa és Open Source el que significa que és una interfície on el contingut que es crea es pot trobar de manera gratuïta i ser redistribuït lliurement per a que tothom ho pugui utilitzar. El benefici per part de l'empresa de Qt es dona quan vols treure profit de les aplicacions que generis amb aquest, que hauràs de pagar-ne una part al servei.

Aquest programa té una interfície força visual i resulta entenedora si ja has programat algun tipus de codi en anterioritat. Es programa amb Java Script i té diferents eines per a poder dissenyar qualsevol aplicació de la manera que sigui més còmode. Pel que fa a la creació de videojocs, també té alguna

llibreria opcional la qual pots instal·lar en la que els gràfics presenten un paper molt important i ja hi ha funcions i operacions que et faciliten la feina de programació per aquest tipus d'aplicacions.

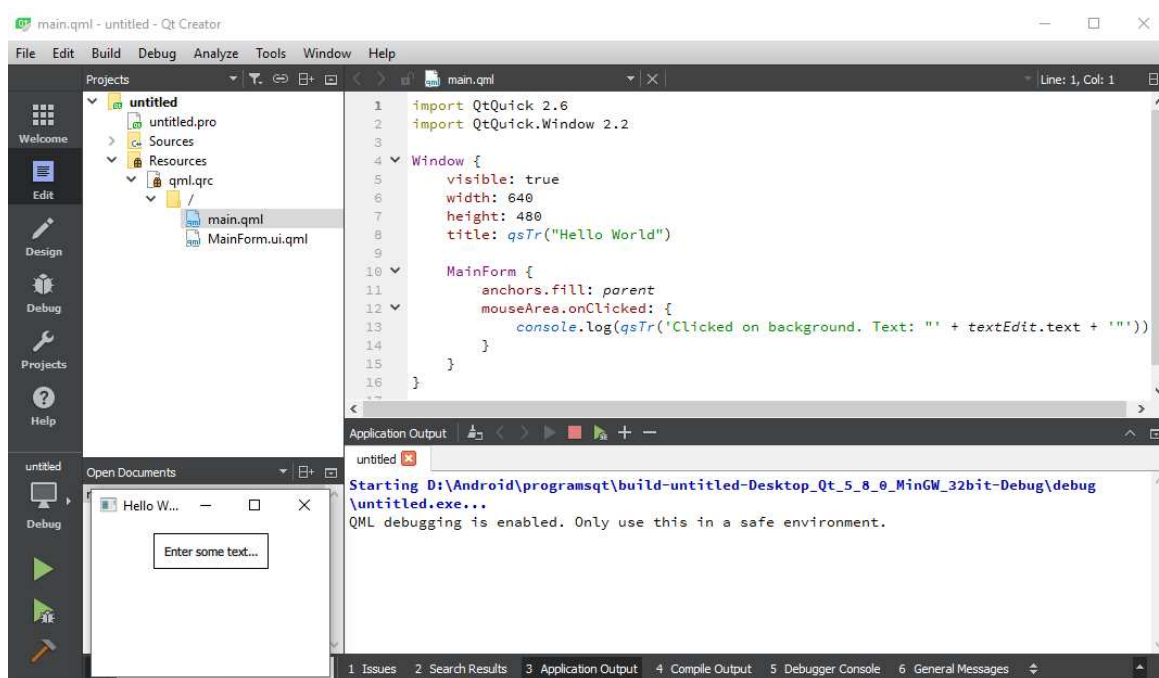


Figura 3.17. Interfície del programa Qt Creator.

El programa es pot baixar de forma gratuïta des de la pàgina principal.

Si el sistema operatiu del ordinador es Windows, farà falta tenir descarregat un conjunt de components que es troben en una llibreria de Visual C++ que són necessaris quan es vol executar un programa en C, C++ i C++/CLI. Visual C++ és un entorn de desenvolupament integrat (IDE) de Microsoft.

La llibreria es pot descarregar gratuïtament des del següent enllaç:

<https://www.microsoft.com/es-es/download/details.aspx?id=40784>

Per a poder portar una aplicació al sistema operatiu Android, una tasca força complexa, caldrà baixar tres paquets d'internet. El primer d'ells és el Java Development Kit o JDK. El JDK més nou que hi ha al mercat és la versió 8, i com ja s'ha dit, serveix per a fer aplicacions i components utilitzant el llenguatge de programació Java. El fitxer que ens interessa de tots els que surten a la pàgina tindrà un aspecte semblant a aquest *jdk-8u77-windows-x64.exe*, on el 8 serà la versió, Windows el sistema operatiu del ordinador, i x64 els bits del sistema operatiu.

El segon component es diu Android SDK o Software development kit, que com ve diu el nom, és ek conjunt d'eines per a poder desenvolupar contingut digital com aplicacions o jocs per a una plataforma, en aquest cas, Android.

El kit es pot trobar a la pàgina oficial de Android seguint aquest enllaç que porta a Android Studio, Android Studio vindria a ser el programa per a fer funcionar SDK, el Qt també ho és pel que no interessa aquest software, el que interessa són les eines i es troben al final de la mateixa pàgina.

Un cop descarregat, s'haurà de descomprimir i posar a una carpeta sense espais, recomanant la C:/Android/SDK.

Dins de la carpeta tools executem el android.bat i se'ns obrirà una pantalla similar a aquesta:

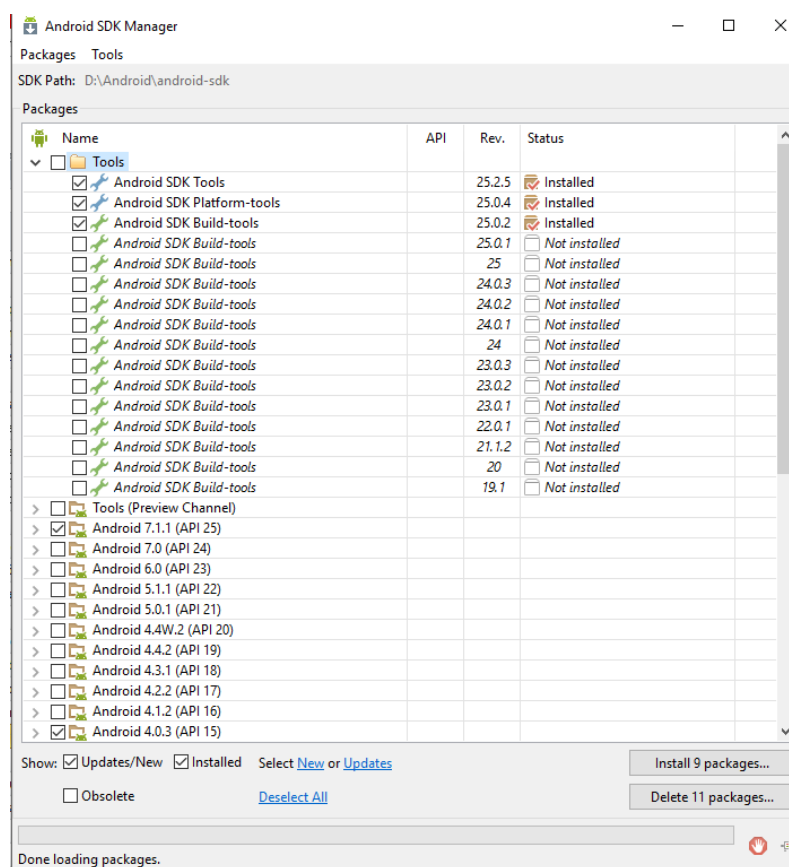


Figura 3.18. Finestra de descarregar dels components SDK d'Android.

En aquesta finestra s'ha d'instal·lar els tres primers components de Tools com es mostra a la imatge i una versió d'Android amb la qual funcionarà la nostra aplicació.

Aquí hi ha el dubte de quina versió escollir. Cada una d'elles té una API diferent, per exemple la versió 7.1, la més nova té API 25, i amb ella es podran utilitzar les funcions que s'han incorporat més

recentment a Android, multi pantalla, realitat virtual, etc. Però tots els usuaris amb una versió anterior a aquesta en el seu dispositiu mòbil no la podran fer servir. El que interessa doncs és agafar la que tingui una API més baixa sense haver de perdre característiques que podrien ser indispensables per a la nostra app. De fet, hi ha aplicacions que es programen amb diferents APIs per a que tots els usuaris puguin descarregar-se-la i els que tinguin un dispositiu més modern, puguin gaudir de millors prestacions en ella.

Pel que fa la nostra aplicació, tot i que encara no s'han definit tots els punts i això es farà en el seu apartat corresponent, podem pensar que per exemple el GPS no es va poder fer servir fins a la versió 2 com, pel que ja no agafaríem la 1. Per altra banda, actualment les versions que tenen més Smartphones estan compreses entre la 4.4 i la 6.0, pel que agafar una versió com la 2 potser no seria necessari i estaria limitant en un futur alhora de fer l'aplicació. S'ha trobat adequat agafar la 4.4 així la gran majoria de dispositius podran accedir a ella i la versió 4 ja és una amb moltes prestacions. A més a més, ara mateix la versió del meu dispositiu mòbil (bq Aquaris E5 HD) és la 4.4 KitKat, agafar una més alta suposaria que no podria provar l'aplicació en el meu dispositiu mòbil.

Un cop acabada la descàrrega de tots aquests components, que tarda molt, com a una hora si només és una versió la que s'ha descarregat, caldrà descarregar-se l'últim component per a Android, Android NDK o Native Development Kit que serveix per a que codi i biblioteques escrites en C, C++ i altres llenguatges puguin ser compilades en ARM, MIPS o x86, codi natiu d'Android com ja vam veure. Aquest kit no és tant costós de descarregar com el SDK però també s'ha de baixar a una carpeta sense espais en el nom aconsellant ser C/Android/NDK.

Per últim farà falta descarregar-se un conjunt de llibreries anomenades Open SSL que ajuden al sistema a implementar el Secure Sockets Layer (SSL), així com altres protocols relacionats amb la seguretat, com el Transport Layer Security (TLS). Es pot trobar de forma gratuïta al següent enllaç.

Amb tot això ja hauríem de ser capaços de crear una aplicació i a més, pujar-la a qualsevol dispositiu amb sistema operatiu Android de versió igual o superior a la del SDK. Ja s'ha vist que els requisits no són pocs i, de fet, poden sorgir molts errors en algun pas de tots ells.

3.5.3. Exemples i aplicacions de Qt Creator

Entendre el funcionament i la metodologia de treball del programa Qt Creator no és una tasca senzilla. Com a prerequisit per la programació de l'app s'ha hagut de provar i entendre molts projectes tutorials. A continuació s'exposarà i explicarà per sobre els que més importància han tingut per a mi. Cal dir que tots ells s'han anat treballant a classe.

3.5.3.1. Exemple 1. Conceptes, elements i funcions bàsiques.

En aquest tutorial es veu l'esquema del programa. Quan s'executa el que es veu és la finestra principal, anomenada Window, a la qual s'hi poden afegir diferents elements com ara cossos geomètrics com un rectangle, o texts escrits. A més a més, es pot incorporar un espai designat amb el nom d'àrea de ratolí i aquest es pot programar per a que es faci alguna acció quan es polsi a sobre d'ell. Per norma general, un click és l'accionador d'una sola àrea de ratolí (a partir d'ara passarem a anomenar amb el terme en anglès Mouse Area tal i com fa el programa) pel que si hi ha dues solapades, la de més a sobre serà la que rebí l'acció.

Per últim es jugarà també amb l'alçada i l'amplada de la finestra principal i les comandes:

onHeightChanged

onWidthChanged

Aquestes serveixen per a que el moment en què el pare (anomenem pare al element que conté el codi escrit) canviï les seves dimensions s'executin aquestes ordres. Ara s'anirà partint el codi veient què fa cada operació.

```
import QtQuick 2.3
import QtQuick.Window 2.2

Window {
    id: window1
    visible: true
    onHeightChanged: console.log("Nova altura = " + height)
    onWidthChanged: console.log("Nova amplada = " + width)
```

Els import són diferents llibreries del programa, la funció Window és la finestra, tot el que es trobi dins dels parèntesis de la funció es mostrarà al moment de executar el programa. Id serveix per a donar-li un nom a, en aquest cas, la finestra per a després poder referir-nos a ella fora de la funció. visible: true farà que la finestra es mostri quan l'executem per a poder ser vista. Els dos últims paràmetres, mostren un missatge per la consola de Qt amb un missatge que copiarà dins de cometes juntament amb una variable, l'alçada o l'amplada del pare en píxels (en aquest cas la finestra) si es canvia alguna d'aquestes variables.

```
MouseArea {
    anchors.fill: parent
    onClicked: { Qt.quit() }
}
```

anchors.fill és una característica que se li pot afegir a un element per a que ocupi una zona específica. En aquest cas ocuparà el parent o el pare, és a dir, la finestra ja que la Mouse Area es troba dins d'ella. onClicked és l'accionador de quan es polsa l'àrea, i aquí simplement es fa que es tanqui l'app amb la operació Qt.quit()

```
Rectangle {
    id: rectangle1
    anchors.centerIn: parent
    width: parent.width/2
    height: parent.height/2
    color: "#0c0ce0"
```

Es declara un rectangle i dins ell se li especifiquen les característiques. anchors.centerIn fa centrar l'element, en aquest cas, al pare que és la finestra pel que es mogui com es mogui aquesta, el rectangle sempre es trobarà centrat al mig. Ja es veu fàcilment que l'alçada i amplada del rectangle seran la meitat de la de la finestra i que el color d'ell ve marcat per un codi estàndard hexadecimal RGB on el 00 correspon a no tenir color i el FF a tenir-ho al màxim; els dos primers són d'escala de vermells els segons a escala de verds i els dos últims a escala de blaus (per tant el codi 000000 donarà color negre, el FFFFFFFF blanc i les combinacions donaran tot un ventall de colors mescla molt ampli.

```
Text {
    text: "Ample = " + parent.width + " alt = " +
    parent.height
    anchors.centerIn: parent
}
}
```

Aquí definim un text amb el valor que es veu després de la seva característica text. Estarà centrat al rectangle ja que ara el pare no és la finestra sinó el rectangle anterior si seguim els parèntesis.

Si s'executa aquest programa el resultat és el esperat i es mostra a continuació.

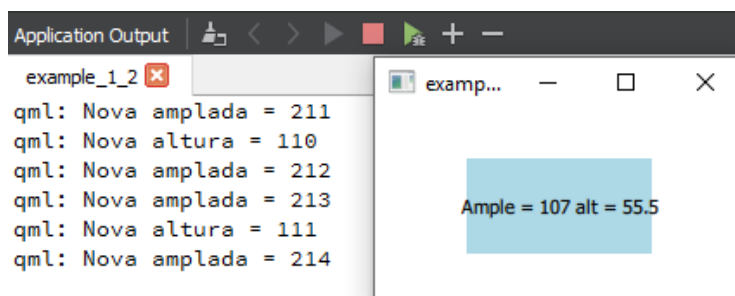


Figura 3.19. Consola i finestra del programa realitzat.

3.5.3.2. Exemple 2. Files, columnes i Imatges .

En aquest exemple es farà servir l'element `columns` i inclourem imatges d'internet al projecte.

```
import QtQuick 2.3
import QtQuick.Window 2.2

Window {
    visible: true
    width: 400
    height: 110

    Rectangle {
        width: 320
        height: 110
        color: "lightblue"
        anchors.centerIn: parent

        Row {
            anchors.centerIn: parent
            spacing: 25
```

`Row` significa fila en anglès. Amb aquest element, tots els elements que es trobin dins d'aquest es posaran en fila amb un espaiat de 25 píxels i la fila estarà centrada al rectangle creat anteriorment, el `parent`. Si es volgués que fos una columna i no pas una fila d'elements, s'hauria de canviar `Row` per `Column`.

```
        Image {
            width: 100
            height: 100
            source: "imgs/blackboard.png"
        }
    }
}
```

L'element `image` posa una imatge. Aquestes d'aquí es trobaran en fila al estar dins de la `Row`, i a la propietat `source` és on s'ha de definir on ha de buscar Qt per a trobar la imatge. Per a posar-ho i que no doni problemes, s'ha d'anar a la carpeta que inclou el programa (`main.qml`), botó dret i `Add existing directory`. Busquem el directori on hi ha les imatges i acceptem. Ja es pot cridar correctament a la `source`, en aquest cas la carpeta `imgs` conté tres imatges extretes d'internet.

```
        Image {
            width: 100
            height: 100
            source: "imgs/house.png"
            MouseArea { anchors.fill: parent
                onClicked: parent.source="imgs/danger.png"
            }
        }
    }
}
```

Aquesta última se li ha incorporat una `MouseArea` per a que canviï la `source` del pare d'aquesta, és a dir la `source` de la imatge canviï per a una altra. Així quan es premi canviarà la imatge per una altra.

El resultat és l'esperat i l'efecte de canvi d'imatge funciona bé.



Figura 3.20. Finestra del programa 3.

3.5.3.3. Exemple 3. Programació de llistes.

Per a fer llistes d'objectes seran necessàries tres funcions: `ListModel`, aquesta tindrà tots els elements de la llista i les característiques de cadascun d'ells; `Component Delegate`, contindrà l'estructura de la llista en sí i `ListView` que mostrarà la llista dins la finestra. En aquest exemple es simularà una llista de contactes amb noms al atzar.

```
ListModel {
    id: modelContactes
    ListElement {nom: "Armand"; telefon: 12345}
    ListElement {nom: "Cristian"; telefon: 67347347}
    ListElement {nom: "Luning"; telefon: 98796421}
    ListElement {nom: "David"; telefon: 341346}
    ListElement {nom: "Marc"; telefon: 4321432}
}
```

`ListModel` que conté un id i els elements que voldrem afegir a la llista. En aquest cas cada element té un nom i un telèfon. Cal tenir en compte que tots els elements han de tenir els mateixos subelements.

Es fa un component amb el que és el delegate de la llista. Les propietats de la llista seran un rectangle amb un color gradient entre el blau i verd definit al final del programa, i columnes de text de poca altura i amplada igual a la finestra amb índex (que ja vam veure que era un número identificatiu), nom i telèfon definits al `ListModel` per a que canviï cada vegada que ho faci els contactes.

```
ListView {
    anchors.fill: parent
    model: modelContactes
    delegate: delegateContactes
}
```

`ListView` amb el que se li defineix el seu model i delegate i se li ha afegit un header amb un text i un footer que són la capçalera i el final de la llista.

```
Gradient {  
    id: gradientLlista  
    GradientStop { position: 0.0; color: "lightblue"}  
    GradientStop { position: 0.66; color: "lightgreen"}  
}
```

El gradient utilitzat en el color del rectangle del delegate. La propietat utilitzada, GradientStop declara quan es fa el canvi entre colors. El resultat és el següent:

3.5.3.4. Exemple 4. Agafar dades d'internet

Amb aquest exemple s'accedirà a una pàgina web de lliure API per a rebre el temps que fa a una localització en concret a temps real. El programa és complex, tot el que té a veure amb internet segueix uns protocols que ja s'han vist i que el servidor et respongui correctament requereix d'un conjunt de passos força complicats. Tot i així aquest programa pot ser-nos molt útil en el disseny de la aplicació.

Primer es farà una llista d'elements com en el del exercici anterior en els que hi haurà longitud, latitud, el temps que faci i la temperatura que hi hagi.

```
Component.onCompleted: {  
    request("http://api.openweathermap.org/data/2.5/weather?" +  
        "q=Barcelona,es"+  
        "&APPID=192e9e332e0ac530bc7e966ecc3199c8",  
        procesaResposta,  
        "GET", "")  
}
```

En aquesta funció es demana al servidor de openweathermap.org una informació específica, la que es troba allotjada al link que es veu. A més, hi ha un apartat que és per a posar la clau de la API individual que et donen quan et registres a la seva pàgina web. Farà falta també crear una funció que ho demani al servidor i una que en processi la resposta. Per últim es posa què es vol fer al servidor, en aquest cas com és agafar dades, farem un GET.

```
function request(url, callback, comando, parametros){  
    var req = new XMLHttpRequest()  
    req.onreadystatechange = creaCallBack(callback, req)  
    req.open(comando, url, true)  
    req.setRequestHeader("Content-Type", "application/json")  
    req.send(parametros)  
}  
function creaCallBack(funcion, peticion){  
    return function() { funcion(peticion) }  
}
```

Aquestes dues funcions són les necessàries per a demanar a la pàgina web les dades. De manera general, es fa una demanda pel protocol http.

```
function procesaResposta(msg) {
  console.log(msg.readyState)
  if(msg.readyState == 4){
    var obj = JSON.parse(msg.responseText)
    var lon = obj.coord.lon
    var lat = obj.coord.lat
    var temps = obj.weather[0].main
    var temperatura = obj.main.temp
    console.log(lon, lat, temps, temperatura)
    modelTemps.append( { "lon":lon, "lat":lat, "temps":temps,
      "temperatura":Math.round(temperatura-273) } )
  }
  console.log(msg.responseText)
}
```

La funció `procesaResposta` agafa la resposta del servidor que té aquest aspecte:



Figura 3.21. Informació en API del temps a Barcelona.

El primer que fa és esperar a que li arribi un codi 4 del servidor. Això voldrà dir que la comunicació entre l'ordinador i el servidor ha estat satisfactòria i que ha superat els controls per a que la informació que s'ha enviat ha estat la correcta. Aleshores toca dividir la informació o parsejar-la com es diu en informàtica per a poder agafar les parts que ens interessin. Les coordenades es troben dins de el grup `coord` pel que les agafarem d'allà, el temps es troba dins de la llista `weather` dins de `main`. El [0] significa que dins la llista `weather` es troba a la posició 0. Per últim la temperatura es troba dins de `main` a la variable `temp`. Al final es posen les variables que contenen la informació accessibles per a la llista recordant que la temperatura es troba en graus kelvin i s'haurà de restar 273 a més d'arrodonir-ho per evitar deu decimals que en temperatura no són gaire importants.

El resultat queda així:

```
Starting D:\projectqt\build-tempss-Desktop_Qt_5_8_0_MinGW_32bit-Debug\debug\tempss.exe...
QML debugging is enabled. Only use this in a safe environment.
qml: 1
qml:
qml: 2
qml:
qml: 3
qml: {"coord":{"lon":2.16,"lat":41.39},"weather":[{"id":801,"main":"Clouds","description":
293.65,"pressure":1020,"humidity":40,"temp_min":292.15,"temp_max":295.15},"visibility":10
1492000200,"sys":{"type":1,"id":5485,"message":0.0022,"country":"ES","sunrise":1491974133
200}
qml: 4
qml: 2.16 41.39 Clouds 293.65
qml: {"coord":{"lon":2.16,"lat":41.39},"weather":[{"id":801,"main":"Clouds","description":"few clouds","icon":"02d"}],"base":"statf
293.65,"pressure":1020,"humidity":40,"temp_min":292.15,"temp_max":295.15},"visibility":10000,"wind":{"speed":4.6,"deg":150},"clouds
1492000200,"sys":{"type":1,"id":5485,"message":0.0022,"country":"ES","sunrise":1491974133,"sunset":1492021743,"id":3128760,"name":
200}
```

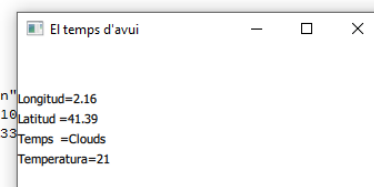


Figura 3.22. Finestra del programa i informació rebuda del servidor.

4. Estudi de mercat

Si bé no es pretén la posada en venda d'un producte, pel que la necessitat de fer un estudi per a veure si és o no viable comercialment no és gaire important, si que pot ser interessant aquest estudi tant per si en algun moment sí que es volgués dur a terme com per a veure perquè els compradors estarien interessats en ell.

Per altra banda es veuran tots els models que puguin ser semblants al mercat actualment, per a analitzar com de competitiu, diferent i innovador és el producte dissenyat.

4.1. Anàlisi del consumidor

Durant el desenvolupament del projecte, s'ha anat fent una enquesta per les xarxes socials per a veure què opinava la gent del dispositiu. El problema de no haver fet un estudi ben fet és que les persones em coneixen de manera personal i sempre és més difícil que valorin negativament un producte. Tot i així la premissa era ser tant transparent com fos possible.

L'enquesta tenia tres preguntes i s'ha classificat en sectors d'edat. De 15 a 18 anys, les persones que la van respondre foren 6, de 18 a 30 anys la van respondre 25 persones i de més de 30 només 8.

Les preguntes eren les següents:

- Pregunta 1. Del 0 al 10, especifiqui què opina vostè d'un dispositiu que sigui capaç d'enviar la seva posició i a través d'una aplicació saber on es troba en tot moment. (Exemple, localització de cotxes, gent gran o nens petits, presos amb llibertat condicional, qualsevol cosa que necessiti ser rastrejada).
- Pregunta 2. Especifiqui quin preu li posaria al dispositiu incloent un usuari privat per a fer funcionar la aplicació.
- Pregunta 3. Especifiqui si ara mateix estaria disposat a comprar aquest dispositiu amb el preu al que vostè ha declarat a la pregunta anterior.

A continuació es mostren els resultats obtinguts en una taula. Ja es té en compte que l'estudi hauria de tenir més enquestats per a poder ser més fiable, però ja serveix per a fer-se una idea del que opina la gent del projecte.

	15-18 anys	18-30 anys	30-50 anys
Pregunta 1:	7.2	6.6	9
Pregunta 2:	27	34	120
Pregunta 3:	0	0.2	0.375

Taula 4.1. Taula amb els resultats de l'enquesta.

El resultat més fiable ha estat del sector d'entre 18 i 30 anys, que han respost de mitjana que la idea era bona, amb un 6.6 de mitjana. Aquest sector ho veu com un producte o aparell divertit però li veuen la pega que has de tenir coses de valor per a poder ser d'utilitat. Només 5 persones l'haguessin comprat al moment. La gent més jove opina força semblant als anteriors, amb que la idea és força bona però cap d'ells va voler comprar el dispositiu. Bé ja es veu que el sector al que va dirigit aquest aparell no és per a la gent jove. Per últim i més sorprenent, el sector adult, tots ells amb fills li han donat una valoració molt bona i un preu molt més elevat que els anteriors. Opinen que si tinguessin un cotxe esportiu no els importaria pagar xifres al voltant dels 200€ per a localitzar el cotxe si l'aparell funciona bé i és dissimulat. Per altre banda també opinen que si n'haguessin tingut la oportunitat quan els fills eren petits, no haguessin dubtat a incloure el dispositiu a una butxaca del fill en cas que es perdés. Un matrimoni em va comentar que per a la seva mare de 75 anys amb problemes mentals tampoc dubtarien a comprar-lo. 3 dels 8 enquestats en aquest sector el comprarien sense pensar-s'ho dues vegades.

Bé, totes aquestes dades donen suport més aviat positiu del projecte pel que la viabilitat per part dels consumidors sembla ser bona. Ara es veurà com es troba el mercat, la competència i a quin preu van tots els productes similars.

4.2. Anàlisi del mercat actual

És lògic pensar que un producte com el que es vol dissenyar ja existeixi en el mercat. Anem a veure com es troba el sector, si existeix algun exactament calcat al proposat, quins preus ronden i a qui va enfocat.

Fent una mica de cerca, grans pàgines de venda online ofereixen productes similars però de connexió Bluetooth. Aquests tenen aplicacions com per pèrdua de les claus, saber on has aparcat el cotxe i altres on la distància entre emissor i receptor sigui de pocs metres. Com es pot pensar, aquests productes estan més limitats i no aporten tota la funcionalitat que donaria un connectat a la xarxa.

Per punts a favor, un preu més reduït, en torn al 10 euros, menys consum de bateria i menys espai. Tot i així no són aquests els que interessin.

Un dels productes que s'està posant molt de moda i incorpora (algun d'ells) la funcionalitat de GPS tracker, són els SmartWatches o rellotges intel·ligents. D'aspecte similar a un rellotge digital amb un receptor GPS i connexió a la xarxa a través de GSM, aquests dispositius són capaços de rastrejar la posició i enviar-la a un dels contactes amb un botó d'emergència. Necessiten d'una targeta SIM i són en definitiva, un bon recurs per a persones grans i nens, però no per a objectes.



Figura 4.1. Rellotge amb GPS i GSM (Font: Amazon [35]).

Si ens centrem a buscar un producte més similar al del projecte, hi ha un que destaca per la resta. Un dispositiu de la marca de SnowFox, de mida compacte i entrada de carga micro-USB que és capaç d'enviar coordenades a temps real a més a més admet trucades telefòniques que poden ser despenjades amb un botó que incorpora el dispositiu. Aquest mateix botó permet enviar un SMS a un telèfon determinat per si volgués que el truquessin per exemple. El preu de venda al públic és de 149€ més 10€ al més per a mantenir la comunicació 3G que per cert, no té lloc per cap targeta SIM, amb aquests deu euros, la companyia s'encarrega d'establir la connexió adient. Té també una aplicació que et permet veure si hi ha cobertura GPS, on es troba el dispositiu i un historial de llocs on ha estat visualitzant Google Maps com a servei de mapes. En definitiva un molt bon producte.



Figura 4.2. Dispositiu de rastrejament per a persones (Font: Snowfox [36]).

S'han vist dos productes i cap d'ells està enfocat al rastrejament d'objectes com vehicles tot i que aquest últim podria fer-ne la funció. Però això no vol dir que no existeixin, un d'ells, clònic al del projecte està en venta a Amazon per un preu de 44€. Amb aplicació, xarxa 2G i targeta SIM no inclosa però necessària, T8 Mini GPS tracker és el millor candidat per a fer una valoració de mercat del producte.

A la seva descripció es pot veure que es troba enfocat a mascotes, persones, vehicles i equipatge per a no perdre res durant el trajecte.



Figura 4.3. Dispositiu de rastrejament T8 (Font: Amazon [37]).

En conclusió, tot i que sembli que el mercat no està molt explotat, realment tampoc hi ha molta demanda del producte, imagino que pel fet que no es tracta de dispositius econòmics. Tot i així no és motiu per a dissenyar un similar a tots aquests, però amb menys prestacions ni un acabat de venta al públic. El que s'intentarà aconseguir però, serà una autonomia més elevada, tots els aparells vistos fins ara, tenien una bateria molt limitada, aquest últim de tant sols 450mA/h. Vist que el mercat no contempla un aparell capaç de ser deixat en un lloc per més de quatre dies, aquest serà un dels objectius diferenciadors del mercat actual.

5. Estat de l'art

En aquest apartat es pretén buscar com es troba actualment la tecnologia utilitzada en el treball, és a dir, es farà recerca de com ha avançat en els darrers anys, quina és la que predomina en el sector.

S'ha dividit en diferents subapartats ja que el projecte consta de diferents blocs independents.

5.1. Receptor GPS

Actualment podem trobar receptors fixes i portàtils. Parlem dels fixes els que són instal·lats en avions, trens cotxes o qualsevol vehicle i els portàtils són aquells de mida petita que no estan ja muntats en un equip o vehicle extern. En aquest projecte trobem la necessitat d'utilitzar-ne un de portàtil que a més a més es puguin tractar les dades rebudes per a poder processar-les per un microprocessador.

Una característica que s'ha anat polint al llarg dels anys és la sofisticació d'aquest component. Existeixen receptors molt petits capaços de ser afegits a un petit Smartphone i de més grans per a vehicles per exemple. La diferència entre la grandària és, en general que la antena de la que disposen tots els receptors GPS és un element indispensable per a rebre senyal, i en un dispositiu petit, aquesta haurà de ser més curta, senzilla i vulnerable a les interferències tant del microprocessador que hi ha darrera com del propi mòdul receptor. En canvi amb un de més gran, pots optar per a una millor antena.

Per altra banda, la velocitat de refresc de dades també ha suposat una millora en el component, no tant per la pròpia velocitat en sí, sinó per el preu al qual es feia pagar per a velocitats més elevades. Aquesta característica ronda per a la majoria de dispositius 1 Hz, és a dir refresca les dades un cop per segon. Normalment tampoc interessa massa que la posició la qual es trobi el receptor s'actualitzi tant ràpid; caminant, conduint entre d'altres aquest ritme de refresc n'és suficient. Si es parla però en aviació, UAVs i altres vehicles aeronaútics veloços, aquí ja sí que potser faria falta 5 Hz o fins 10 Hz de velocitat de refresc.

Cada model de receptor té un cert nombre de canals. Quants més canals tingui, més ràpid serà de mitjana el receptor en trobar les senyals dels satèl·lits i per tant, de localitzar-se.

Només fa falta buscar freqüències quan, o bé s'enrega el dispositiu per primer cop, en aquest cas es necessita la cerca de quatre satèl·lits i és el que rep el nom de Time-To-First-Fix (TTFF) o bé quan es perd la senyal d'un d'ells ja sigui perquè ja no és apreciable i s'ha difós per l'horitzó, o un edifici per

exemple en bloqueja les ones. En aquest darrer cas la localització seria possible sense saber-ne l'altura fins a trobar un altre satèl·lit a l'abast, moment en el qual es recuperaria aquesta dada [38].

Si ve la tecnologia avança i cada cop hi ha més receptors amb més canals a un preu més raonable, fins a quin punt resulta òptim tenir-ne un cert nombre? I ja que estem posats, si el nombre màxim de satèl·lits que es poden arribar a veure en un mateix moment en un punt de la Terra són 12, i la constel·lació actual en contempla 24, perquè hi ha receptors de més de 50 canals?

La resposta és la següent: Si per exemple tinguéssim un receptor de 12 canals, aquest podria utilitzar quatre de les senyals que trobés, trobar també la senyal de dos o tres en el cas que es perdés la senyal d'algun utilitzat en aquell moment i fins i tot tenir algun més buscant continuadament per així tenir una imatge 3D continuada de la posició. Tot depèn de la programació. Amb 12 canals, la mitjana de temps que fa falta per a trobar les coordenades varia de mitjana entre 1 i 3 minuts. I on de veritat s'aprecien el nombre de canals és en el moment d'engegar el dispositiu i fan falta 4 senyals. En aquest moment tots ells es posen a buscar fins a trobar-ne una pel que tot i que tinguis 50 canals, pot ser que més d'un estigui cercant el mateix PRN. Per tant, quants més canals, més rapidesa en la localització inicial.

També hi ha altres factor a tenir en compte, com el consum de l'aparell. Si el chip de captura de senyals ha d'estar constantment buscant les ones dels satèl·lits, el consum augmentarà considerablement si només fa falta fer-ho 10 ms cada segon. Per altra banda, tot i que només fan falta 4 canals per a definir les coordenades, molts fabricants utilitzen més per a definir el seu algoritme i tenir una millor precisió [1], [38].

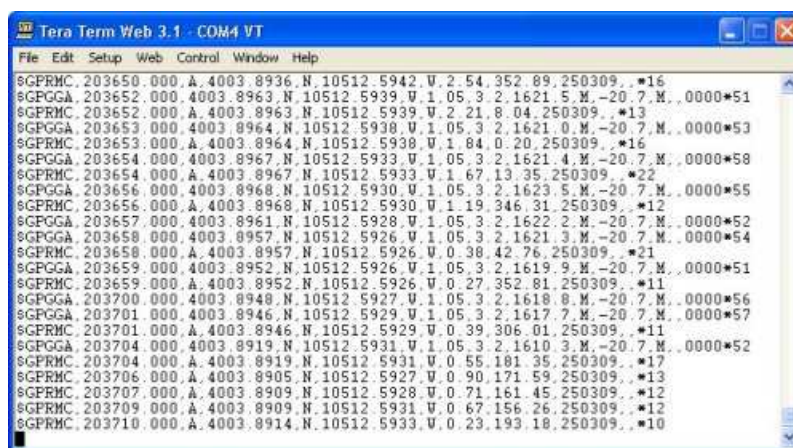


Figura 5.1. Imatge de les dades rebudes per a un receptor GPS. (Font: spark fun electronics [38])

5.1.1. L'avanç de la tecnologia: DGPS i RTK

Les sigles DGPS corresponen a Differential GPS o el que és en català, GPS diferencial.

Aquesta tecnologia GPS preveu d'una millora molt notable en la precisió que es rep de la localització dels satèl·lits. L'error passa a ser estipulat de ± 15 m a ± 0.1 m en les millors implementacions.

Aquesta millora s'aconsegueix amb xarxes fixes d'estacions terrestres que emeten la diferència entre la seva posició indicada pels satèl·lits i la posició precisa real per així els receptors derivar-ne també diferència. La teoria doncs d'aquesta tecnologia rau en que dos receptors GPS relativament propers experimentaran errors similars de precisió degudes a interferències meteorològiques. Si s'instal·len suficients estacions que sàpiguin exactament les seves coordenades i les comparin amb les rebudes pels satèl·lits, podran transmetre aquest error a altres receptors propers per a poder evadir-los eficaçment. La comunicació pot donar-se per exemple per ones de ràdio [39].

Actualment la majoria de receptors tenen aquesta disponibilitat, i l'utilitzat no en serà una excepció.

Les sigles RTK corresponen a Real Time Kinematic que es ve a traduir com navegació cinètica en temps real. Es tracta d'una tècnica de GPS diferencial que necessita com ja s'ha explicat, d'una estació de referència terrestre.

El funcionament d'aquesta tecnologia consisteix en la retransmissió per part de l'estació de referència la fase de la senyal portadora que ha rebut a les diverses unitats mòbils que la comparen amb la que han rebut elles. La informació que porta el senyal doncs, no és l'important d'aquesta ona, sinó la seva fase. Amb aquestes dades i alguns mètodes estadístics sofisticats de resolució d'ambigüitats, és possible corregir un gran percentatge d'error [40].

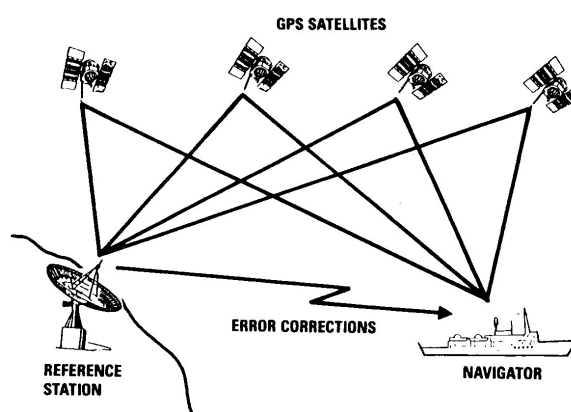


Figura 5.2. Esquema de funcionament del GPS diferencial. (Font: Dronepic [39])

5.2. Cobertura de transmissió de dades

Actualment ens trobem a la quarta generació, els consumidors fa poc que han començat a canviar els seus terminals a la tecnologia 4G tot i que moltes de les xarxes encara no estan preparades per a donar un servei 4G real o el que es denomina *true 4G*.

Tot i això, el servei 3G és àmpliament utilitzat mundialment, de fet més que el 4G però les xifres van més enllà; La tecnologia de segona generació és encara la més utilitzada, sobretot en bona part del món industrial i les comunicacions entre màquines, el que s'anomena Machine to Machine (M2M).

Dades oficials: A nivell mundial, 2G va representar el 53% de les connexions el 2016, amb 3G en el 33% i el 14% de LTE.

Ens trobem en una etapa de transició doncs, on les velles tecnologies ocupen un paper important en les comunicacions tot i tenir alternatives més modernes i sofisticades. AT&T i altres companyies que subvenciona aquestes xarxes, han tingut un debat intern important en aquests últims anys. Mentre que es demanen xarxes LTE 4G millors, part del finançament ha d'anar destinat al manteniment de xarxes 2G i 3G. L'apagat d'aquestes serà evident en els pròxims anys, però quan la societat serà capaç de digerir aquest canvi tant brusc en les xarxes?

Mentre que el 4G es diferencia del 3G per una simple millora tecnològica, no passa el mateix amb el 2G. Aquesta tecnologia es utilitza per exemple en gasolineres, terminals de targetes de crèdit, equips industrials, aparells antics, màquines expenedores (en el moment que detecta la falta de algun producte, aquestes envien un SMS a la companyia perquè vagi a reomplir-la) i d'altres dispositius que no necessiten una alta velocitat de transferència de dades però si un preu baix i una transmissió sense fils. En aquest context es va proposar l'apagat de la xarxa 3G abans de la 2G contra qualsevol expectativa [41].

Però hi ha un altre punt de vista. Si s'elimina la xarxa 3G, en qualsevol moment o lloc on la tecnologia 4G no arribi significarà o bé que no hi haurà possibilitat de connexió i per tant d'enviament de trucades o dades, o es connectarà amb xarxes de segona generació, i passar de 4G a 2G és un canvi massa radical. I aquí es troba el debat, perquè l'empresa no està disposada a subvencionar tres generacions alhora a part que s'ha de començar a pensar amb el 5G.

El dilema ja s'ha resolt i és que en països com EEUU on AT&T ocupa la majoria de mercat, les xarxes 2G seran les primeres en ser eliminades al 01/01/17. A Espanya però, Vodafone assegura la seva xarxa fins el Octubre d'aquest mateix any [42].

No hi haurà més remei que canviar tots els terminals que funcionin amb GSM/GPRS per a altres de tercera i quarta generació, i si pot ser per 4G millor perquè Vodafone vol eliminar el 3G a partir de l'any 2020, en tres anys per enfocar-se completament amb el 4G i el 5G que ja s'estan fent proves i estàndards i un seguit de companyies hi estan treballant.

Tot i així hi ha algunes empreses de telecomunicacions d'altres països europeus que estan discutint apagar primer el 3G que el 2G.

5.2.1. Carrier Agregation i el suposat 4G+

Si bé no s'arriba als estàndards reals de 4G dictats per la ITU (100 Mbps en moviment i 1 Gbps en repòs), ha aparegut una tecnologia que ens apropa a aquests (de fet a Amèrica del Nord AT&T vol arribar-hi aquest mateix any). Es tracta de *carrier aggregation* o el que ve a ser traduït, agregació de portadores. Aquesta tecnologia ofereix exitosament pics de velocitats de dades superiors, així com una millor experiència de banda ampla per a terminals de almenys categoria (ue-category) 6.

A Espanya la tecnologia LTE conta amb tres bandes de freqüències (800 MHz, 1800 MHz i 2600 MHz) i cadascuna compte amb unes característiques concretes, la de 800 MHz penetra millor en interiors, les altres són més estables... En funció de la nostra situació, el terminal es connecta en la que presti millors característiques. Amb la carrier aggregation el terminal és capaç de connectar-se a diverses bandes de forma simultània sempre i quan sigui possible per així poder repartir el tràfic de dades, tenir una senyal més estable i el traspàs de xarxes es faria més suau. Gràcies a aquest procés, augmentem l'ample de banda total i podríem arribar als 450 Mbps en un dispositiu categoria 10.

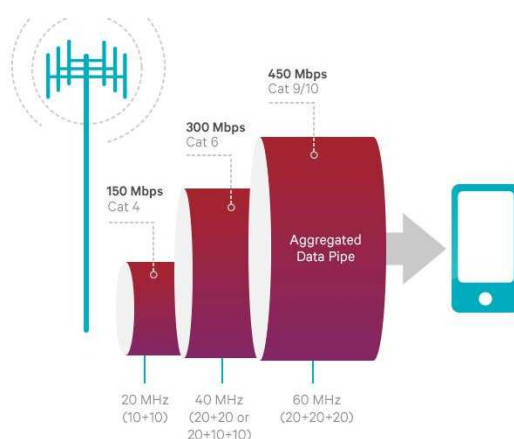


Figura 5.3. Imatge referent a la suma de portadores. (Font: qualcom [43])

A la part inferior de la imatge es veuen 20, 40 i 60 MHz, aquests nombres representen la suma total de l'ample de banda agafat de cada xarxa LTE. Per exemple prenent 20 MHz de la banda de 2600 MHz, 20 MHz de la banda de 1800 MHz i 20 MHz de la banda de 800 MHz [44].

Només hi ha una condició per a poder fer que això sigui factible; que es pugui tenir accés a les tres antenes o almenys a dues d'elles. Tot i que actualment s'ha millorat molt la cobertura d'aquestes xarxes 4G LTE (almenys a Espanya i Catalunya), la adquisició de cobertura de dues o tres antenes és molt escassa, i només es pot adquirir a Madrid Barcelona i Valencia. Un altre detall és que aquesta tecnologia només es pot utilitzar en la baixada de dades, tot i que en un futur serà també possible utilitzar-la en la pujada [45].



Figura 5.4. Mapa de cobertura 4G a Espanya per part de Vodafone. (Font: Vodafone España [45])

Si ve altres operadores de xarxa pròpia també ofereixen el servei de 4G (movistar Orange i Yoigo) , Vodafone és la que més esforç hi ha posat per aconseguir que actualment més del 94% de la població pugui gaudir de la xarxa 4G. Les altres companyies com Simyo, Amena, Lowi etc. No tenen la seva pròpia xarxa de antenes sinó que utilitza les de operadores que si la tenen tot i que no entrarem més en detalls.

5.2.2. El projecte AirGig

AirGig és un projecte de la companyia AT&T que espera posar a la pràctica en aquest any 2017. Es tracta d'una tecnologia revolucionària que algun dia podria oferir accés a internet amb velocitats de diversos gigabits a un baix cost, utilitzant línies de les línies elèctriques.

La companyia té patentades o les està sol·licitant més de 100 patents per a la realització d'aquest projecte que es va donar a conèixer fa pocs mesos. Citant les paraules de la pròpia companyia, "Aquesta tecnologia serà més fàcil d'implementar que la fibra, es pot transmetre per un espectre sense llicència i pot oferir connectivitat mòbil ultra-ràpida a qualsevol llar o dispositiu mòbil. Dissenyem el Projecte AirGig literalment des de zero per crear una opció pràctica i transformativa. Els experiments inicials, que continuen a les instal·lacions a l'aire lliure d'AT&T, han estat positius."

On més impacte podria tenir aquesta tecnologia és en zones rurals o marginades. Amb una línia de mitja tensió es podrien enviar dades a velocitats de diferents gigabits per segon. Sense necessitat de noves torres, sense la necessitat d'enterrar nous cables.

Expliquen també que en cap moment aquesta tecnologia causaria problemes a les empreses de subvencionament d'energia elèctrica, més aviat podria ser-ne d'utilitat i la beneficiaria [46].

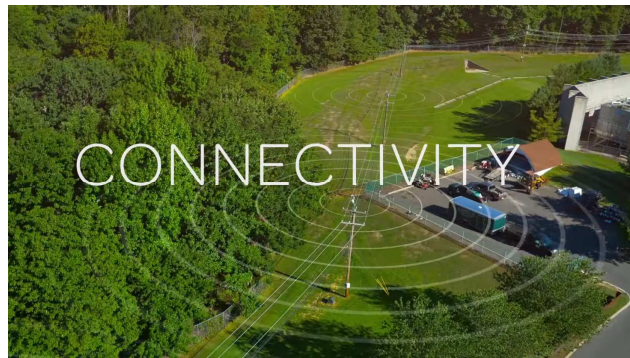


Figura 5.5. Fotografia del vídeo promocional de AirGig. (Font: AT&T [46])

5.3. Servidors

Per a complementar la teoria de fonaments, exposarem quins són actualment els servidors online que més estan utilitzant les pàgines web senzilles com les que requereix el projecte.

Hi ha diverses companyies que ofereixen el servei d'un servidor. Val a dir que hi ha propostes excessivament professionals i en contrapartida, poc econòmiques. El lloguer de servidors dedicats oscil·la entre els 30 i 100€ pels més barats al mes i fins a 700€ els de més prestacions. Amazon és una companyia que ofereix, a part de la venda d'objectes, una gran plataforma de servidors molt ben organitzada sota el nom de Amazon Web Services. El preu varia en funció del tipus de servidor, la seva capacitat, la seva localització, el temps de reserva, si és públic o dedicat, etc. Hi ha petits espais que es poden llogar a 4\$ al mes, el que la quantitat de memòria del servidor és de 0.5Gb. El preu en relació als Gigues sembla força proporcional[47].

Microsoft també compta amb els seus servidors, amb el nom de Microsoft Azure. Comparant preus, aquest servei ofereix un servidor amb una base de dades de fins a 2Gb per 4.20€ al mes. El servei presumeix de ser fiable i segur i amb total llibertat per als desenvolupadors de contingut per a utilitzar les eines que més els hi vagin bé [48].

No podia ser menys Google, qui amb la Google app engine ofereix la possibilitat d'escalar automàticament les aplicacions web i mòbils desenvolupades i assegura ser la més eficaç per al

desenvolupament de servidors que allotgen pàgines web. El preu funciona per Gb usats al mes, amb limitacions de cerques en el servidor, etc [49].

Si bé tots aquests serveis tenen un temps de prova gratuït de un més de mitja, realment cap d'aquest és de gaire interès en aquest projecte. Haurem d'anar a empreses que ofereixin un servei molt més limitat per a evitar costs extrems en tot aquest tram del projecte per evitar haver de pagar una tarifa mensual per a mantenir-lo funcional. La companyia que més possibilitats m'ha semblat tenir en el lloguer d'un servidor és una anomenada Heroku.

Heroku és una empresa que té contractats diversos servidors d'Amazon i la companyia els lloga a parts als seus usuaris amb la possibilitat de crear-te un espai gratuït en ell. Realment la quantitat de particions que té cada servidor és molt elevada per a poder obtenir benefici. La conta gratuïta és molt limitada, el servidor es posa a "dormir" si han passat 30 minuts i no hi ha hagut activitat (ningú ha entrat al servidor), amb un màxim de 5 bancs de proves o com li diuen ells dyno, amb un màxim de 550 hores d'utilització al mes. Mentre no s'hagin gastat, el servidor seguirà operatiu; Aquest temps és el que el servidor està comunicant-se amb alguna aplicació exterior com un altre ordinador amb la seva IP, un mòbil, un mòdul 2G, etc [50].

Realment el servei d'aquesta companyia serà més que suficient per a satisfer les necessitats del projecte i així estalviar-nos una paga mensual, per a aplicacions que necessiten comunicació M2M els servidors d'aquest estil són els més òptims ja que una màquina com pot ser un microprocessador i una app no necessiten de fitxers de grans dimensions, només text pla.

The screenshot shows the Heroku pricing page with a navigation bar at the top containing 'Dynos', 'Databases', 'Add-ons', and 'Support'. The estimated monthly cost is \$0, and there is a 'SIGN UP FOR FREE' button. Below the navigation bar, a headline states 'Dynos are smart, lightweight containers built for modern languages and developer productivity' with a sub-note 'Pay only for what you use, prorated to the second. Change dyno types at any time. Detailed Comparison.'.

Free	Hobby	Professional
<p>Free</p> <p>Ideal for experimenting with cloud applications in a limited sandbox.</p> <p>CORE PLATFORM FEATURES</p> <p>SLEEPS AFTER 30 MINS OF INACTIVITY</p> <p>USES AN ACCOUNT-BASED POOL OF FREE DYNOS</p> <p>CUSTOM DOMAINS</p> <p>512 MB RAM 1 web/1 worker</p> <p>Free</p>	<p>Hobby</p> <p>Perfect for small scale personal projects and hobby apps.</p> <p>CORE PLATFORM FEATURES</p> <p>NEVER SLEEPS</p> <p>FREE SSL & AUTOMATED CERTIFICATE MANAGEMENT FOR CUSTOM DOMAINS</p> <p>APPLICATION METRICS</p> <p>MULTIPLE WORKERS FOR MORE POWERFUL APPS</p> <p>512 MB RAM 10 Process Types</p> <p>\$7 per dyno/month (prorated to the second)</p>	<p>PROFESSIONAL</p> <div> <p>Standard (1X, 2X)</p> <p>Enhanced visibility, performance, and availability for powering your professional applications.</p> <p>ALL HOBBY FEATURES +</p> <p>SIMPLE HORIZONTAL SCALABILITY</p> <p>THRESHOLD ALERTS</p> <p>PREBOOT</p> <p>512MB OR 1GB RAM</p> </div> <div> <p>Performance (M, L)</p> <p>Superior performance when it's most critical for your super scale, high traffic apps.</p> <p>ALL STANDARD FEATURES +</p> <p>MIX WITH STANDARD 1X, 2X DYNOS</p> <p>DEDICATED</p> <p>AUTOSCALING</p> <p>2.5GB OR 14GB RAM</p> </div> <p>Process Types</p> <p>\$25 - \$500 per dyno/month (prorated to the second)</p> <p>Send feedback</p>

Figura 5.6. Pla de pagament dels servidors Heroku. (Font: Heroku[50])

5.4. Aplicacions de plataforma mòbil

És una realitat, aquest tipus de software és avui en dia el més explotat mundialment. Play Store té més de 700.000 aplicacions disponibles per a descarregar-se i qui no té un Smartphone avui en dia? En aquest apartat però ens centrarem a l'altra banda de la moneda, a la creació i no pas utilització de les apps, com es troba actualment el món de creació de software per a dispositius mòbils?

Primer de tot, si es crea una app, després la pots fer compatible amb qualsevol dispositiu mòbil? Perquè no totes les marques i models són del mateix fabricant, i com es troba el mercat, per on val la pena decantar-se si s'ha de fer que ja sembla que té tota la pinta?

5.4.1. Android, iOS, els dos?

El mercat de les apps es troba actualment força dividit. Si ve és cert que moltes aplicacions es troben en les dos plataformes, crear una aplicació senzilla de poc abast no és el mateix que crear-ne una per a fins lucratiu i més reputació. A més a més, l'objectiu d'aquesta app és utilitzar-la per a controlar un dispositiu únic, val la pena treure-la al mercat per a més d'una plataforma?

Primer de tot cal rectificar que no només existeixen dos plataformes, Android i iOS. Cada marca de mòbils elegix la plataforma que vulgui i li interressi per a complementar les aplicacions que tingui el seu producte, Samsung per exemple utilitza la plataforma de Android, plataforma de Google inc. igual que Huawei, HTC, Lenovo, BQ entre d'altres.

Mentre que el telèfon pot ser considerat el hardware i la plataforma el software, la famosa marca Apple va optar en el seu temps en crear un sistema operatiu que funcionés exclusivament amb la plataforma que ells mateixos van crear, iOS i que cap altre hardware pogués utilitzar-la. Aquest aspecte es pot considerar un avantatge, perquè la relació entre hardware i software és molt propera, a més doten el producte amb una exclusivitat que no té la seva competència més directa. Però per altre banda, aquesta exclusivitat també pot ser una limitació. La presència de mercat de Smartphones de diferents marques és molt elevat, i més ara amb la sortida de telèfons de companyies de marca asiàtica més assequibles que els grans models de Samsung o Iphone d'elevat preu. És una realitat, la gran majoria d'aquests fabricants de mòbils trien Android com a sistema operatiu davant la impossibilitat d'utilitzar iOS. I la resta de sistemes no tenen un gran ventall d'aplicacions ni desenvolupadors de contingut digital per a aquestes; Però també iOS és una plataforma sòlida i que val la pena contemplar.



Figura 5.7. Logotip d'Android i d'Apple simulant divisió d'opinions. (Font: worldversus [51])

La pregunta doncs és: A quina plataforma es farà l'aplicació i perquè no en les dues?

Per començar el llenguatge de programació que utilitza per a funcionar Android, és el Java, mentre que el que s'utilitza per a iOS és Objective-C. Com ja es veu a venir, és impossible programar una aplicació i posar-la a les dues plataformes amb el mateix llenguatge. Tot i que hi ha programes que intenten ajuntar els dos llenguatges per a que no es necessiti refer el codi dos vegades, la realitat és que, totes les aplicacions de certa reputació que es troben per les dues plataformes han hagut de ser reescrites en els dos llenguatges. Al cap i a la fi, es tracta de dues maneres d'operar completament diferents.

Amb tot això es pot descartar el utilitzar les dues plataformes simultàniament, ara toca doncs escollir. Deixant de banda temes com que Android domina més quota de mercat o que iOS ingressa més beneficis per part del desenvolupadors, ens centrarem a respondre aquesta pregunta per facilitar i maniobrabilitat alhora de dissenyar i penjar l'aplicació al mercat.

Android ofereix als desenvolupadors una plataforma de desenvolupament oberta i els permet la llibertat d'utilitzar eines de tercers per a la creació d'aplicacions. Això és bo per a la llibertat de moviment pels creadors de contingut. Aquest és un dels punts forts d'aquesta plataforma, que ve amb una impressionant gamma de dispositius mòbils. Apple, per contra, és bastant restrictiva amb les seves pautes de desenvolupament. Al desenvolupador se li dóna un conjunt fix d'eines per desenvolupar aplicacions i no pot fer servir qualsevol cosa fora d'ells. Això a la llarga frena les seves habilitats creatives en gran mesura.

Per altra banda, Els productes d'Apple tenen una forta reputació de seguretat. Els iPhones només poden executar aplicacions que hagin estat aprovades prèviament per Apple, mentre que Android és una plataforma oberta que no requereix més aprovació que superar algun test de seguretat per evitar virus. A més a més, Apple presenta una plataforma més estable, exclusiva per als desenvolupadors d'aplicacions, amb especificació clara de les eines, que defineix tant el seu potencial

i límits. Això fa que sigui molt més fàcil per al desenvolupador poc expert tenir èxit en el desenvolupament de l'app.

Per contra, el sistema operatiu Android molt sovint crea problemes per al desenvolupador aficionat ja que pren molt de temps per aprendre, entendre i dominar. A més aquesta plataforma es troba molt fragmentada [52][53].

Per últim cal mirar quan val publicar una aplicació al mercat de Google i de Apple. Per a poder penjar una aplicació al Play Store cal tenir una compta de desenvolupador, i això costa 25\$ que es paguen un cop i ja et dona dret a tenir el servei per sempre. En el Apple Store, també necessites crear-te una compta de desenvolupador, i en aquesta plataforma costa 99\$ i s'ha de renovar un cop cada any. Bé, hi ha diferència, i per si no fos poc, el dispositiu que actualment estic utilitzant (BQ Aquaris E5) utilitza Android com a sistema operatiu. Dissenyar l'aplicació per a Apple, a part de més car, seria encara més car per haver de comprar un Smartphone d'Apple per a provar-la (i ja se sap que els mòbils d'aquesta companyia no acostumen a ser barats).

En conclusió doncs, es dissenyarà una aplicació mòbil utilitzant la plataforma d'Android.



Figura 5.8. Divertida imatge d'Android i Apple. (Font: theandroid-mania [54])

5.4.2. Programa per a la creació de la app

Android té el seu propi software de disseny d'aplicacions anomenat Android Studio, amb ell ja et venen totes les eines necessàries per a poder utilitzar el programa i que l'aplicació que creïs funcioni si tot s'ha fet com demanen, correctament. Però existeixen altres programes per a fer aplicacions.

- Visual Studio de Microsoft és una de les plataformes amb més renom per la xarxa. Multi plataforma basat però per a fer apps per la marca Apple. Tot i així contempla un emulador per a Android basat en Hyper-V. El servei per a empreses és de pagament però pots baixar-te el pack limitat de forma gratuïta[55].

- KDevelop és també una opció multi plataforma programable a diferents llenguatges incloent C, C++ Java, etc. Lliure i de codi obert. Sembla que té molt bon feedback per part dels seus usuaris[56].
- Qt Creator. A moltes pàgines web nomenat com a millor interfície per al desenvolupament d'aplicacions, fàcil per a usuaris sense experiència i amb moltes funcions i propietats per a poder fer qualsevol cosa que et proposis. A més és per a Android, gratuïta i de codi obert [57].
- Unity. Una poderosa IDE amb molt de renom en la creació especialitzada de jocs de qualsevol tipus de plataforma. També es poden fer aplicacions per suposat però el programa està especialment dissenyat per a fer jocs amb potents gràfics, animacions, etc. La imatge a continuació mostra a la gran quantitat de plataformes a les que es pot dissenyar jocs amb Uniy amb les que es pot destacar Android, iOS, Windows Phone, consoles d'última generació, fins i tot la recentment llançada al mercat Nintendo Switch [58].

Soporte multiplataforma líder de la industria

Reconocido en toda la industria como el motor de juegos multiplataforma, Unity te permite apuntar a más dispositivos más fácilmente. Con Unity, logras el despliegue con un solo clic en toda la gama de plataformas móviles, de VR, escritorio, Web, consola y TV.

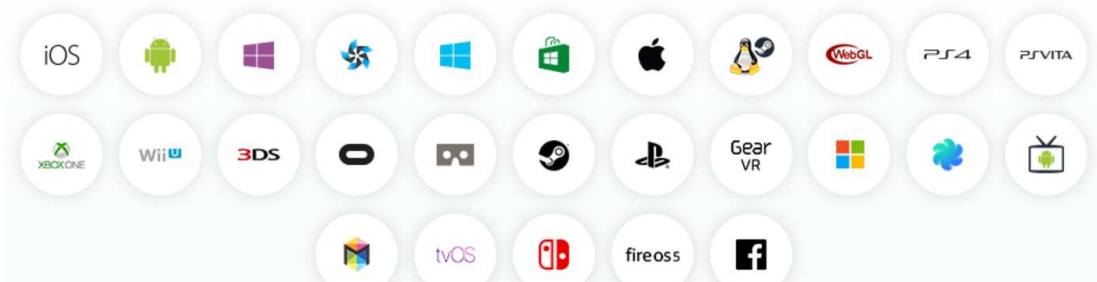


Figura 5.9. Plataformes suportades per Unity en la creació de videojocs. (Font: Unity [58])

Alhora d'escollir, la decisió és més que res per comoditat ja que ja s'ha estat treballant amb la plataforma durant un temps. L'escollida doncs és Qt Creator. Realment és un bon software de desenvolupament integrat que permetrà la creació de l'aplicació en ment.

6. Selecció del material

Gràcies tant als fonaments com a l'estat de l'art descrits anteriorment, la feina de tria dels components es farà més fàcil. En aquest apartat es discutiran les necessitats bàsiques i les característiques mínimes que hauria de complir tot el material que s'adquireixi.

6.1. Mòdul receptor GPS

De receptors GPS n'hi ha de moltes varietats i característiques diferents. A continuació s'exposaran totes les que s'haurien de tenir en compte:

- Mida.
El projecte escollit no presenta cap necessitat de mida increïblement petita. Mentre el dispositiu creat pugui ser transportat fàcilment, no serà un factor important alhora de la tria.
- Velocitat de refresc.
Tampoc és gaire important aquesta variable. Aquest projecte no està destinat a ser sotmès a altes velocitats on la posició podria trobar-se desfasada. Per tant tampoc es limitarà la tria per aquest factor.
- Nombre de canals.
Els necessaris i adients per al projecte. No és condició indispensable que el receptor sigui molt ràpid alhora de buscar la senyals dels satèl·lits. Actualment hi ha receptors amb prou canals com per a tenir un temps d'espera de entre 1 i 3 minuts. Podríem considerar un amb millors prestacions si es donés el cas; sinó, aquest seria el candidat.
- Preu.
Es tracta d'un projecte no subvencionat per ningú més que per l'estudiant, pel que sempre que no resulti un impediment, aquest es trobarà limitat a les necessitats bàsiques i imprescindibles per al projecte.

6.1.1. Mòduls de mercat GPS

- SkyLab SKM53.
Aquest mòdul també anomenat SkyNav està basat en el circuit MT3329 de MediaTek i porta incorporat una antena GPS. Presenta una alta sensibilitat i cobertura en llocs on altres dispositius presentarien problemes.
Aquest receptor compta amb 22 canals de rastreig i 66 d'adquisició és de petites dimensions i el consum és de mitja uns 130 mW i amb pics de 150 mW quan es busca la senyal dels satèl·lits. El preu ronda els 35 euros.



Figura 6.1. Imatge del receptor SkyLab SKM53. (Font: Punto Flotante S.A. [59])

- U-Blox GPS .

Aquesta companyia té diferents models de receptors els quals destaquen la sèrie 6 NEO, llançada al mercat l'any 2011 i que vol destacar per el seu baix consum d'energia gràcies a una administració intel·ligent d'aquesta, un baix cost i de petites dimensions per a satisfer l'estricta demanda per part del mercat actual per a aquests dispositius. Es tracta d'un venedor industrial on els seus productes van destinats principalment a fabricants de dispositius mòbils i a el sector més proper a aquest.

Amb 50 canals i un baix consum, U-Blox és el líder del mercat actualment amb tota la seva gamma de productes.

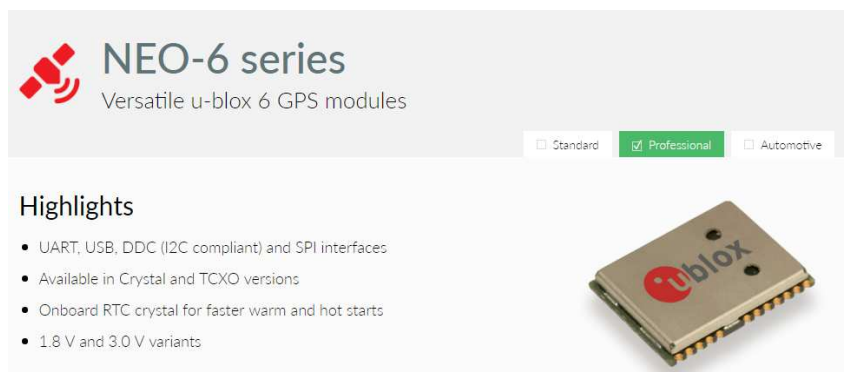


Figura 6.2. Imatge del component extreta de la pàgina oficial de la companyia. (Font: u-blox [60])

- Modul GPS de cooking hacks

Aquesta petita companyia de subministrament de components electrònics ha dissenyat un mòdul receptor GPS sense l'antena inclosa es troba disponible al preu de 42€.

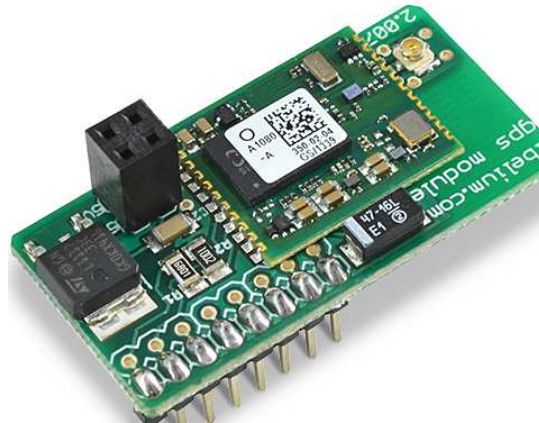


Figura 6.3. Mòdul GPS en qüestió. (Font: cooking hacks [61])

- Adafruit Ultimate GPS breakout.

Un receptor d'una companyia de components i software electrònic de molt bones referències que a més inclou el software i llibreries per a l'ús del seu mòdul. El preu però no és tant competitiu com el de les grans companyies, es troba a 40€.

Porta incorporada l'antena ja soldada i és de dimensions molt petites.

És un bon receptor a tenir en compte per a dissenyar un bon projecte.

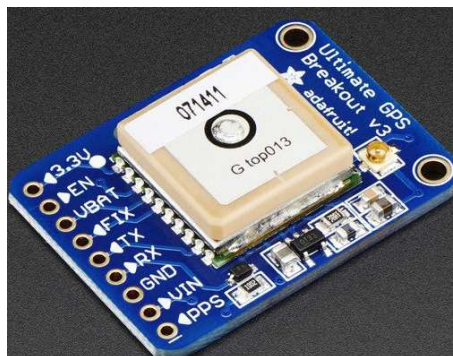


Figura 6.4. Mòdul GPS Adafruit. (Font: Adafruit [62])

- LE910 4G+GPS shield.

Placa increïblement completa per a realitzar qualsevol projecte del que anomenen ja una nova era, *Internet of things* que també és distribuïda per cooking hacks. Aquesta placa consta d'un mòdul capaç de connectar-se a alta velocitat a internet (4G) i alhora disposa també d'un receptor GPS. Porta incloses totes les antenes necessàries i, de fet, aquesta placa resoldria la meitat del treball si no fos per el seu excessiu preu, que volta els 320€ i es ven a pàgines com Amazon a 400€. També haig d'esmentar que un dels objectius és que el cost del prototip final sigui dins el que cap, raonable i competitiu i penso que el projecte es pot fer amb molt

menys pressupost. Tot i així és interessant comentar aquesta placa, i el distribuïdor tornarà a sortir més endavant en el treball.



Figura 6.5. Mòdul LE910 4G+GPS i antenes. (Font: cooking hacks [61])

6.1.2. Elecció del receptor GPS

Tot i que el mòdul SKM53 va resultar molt interessant i una potencial adquisició, la sèrie Neo 6 de U-Blox és molt atractiva i el fet que sigui líder del mercat n'ha fet marcar la diferència. Les característiques són més destacables en aquest últim tot i que això sí, amb menys canals que el seu competidor però amb un preu més atractiu i, per a treure'ns de dubtes, un receptor totalment suficient.

El problema de U-Blox és que es ven el receptor i prou ja que està pensat per a ser distribuït amb quantitats industrials (ja s'ha comentat per exemple per a implementació en dispositius mòbils i altres productes de la mateixa categoria).

Per internet però, es venen mòduls amb aquest receptor on només és necessària una antena i un microprocessador per a fer-lo servir. Una altra alternativa és el disseny de la placa per a poder utilitzar el receptor, el problema més evident però és la quantitat de càrrega que suposaria. Per començar, es comprarà el mòdul ja fet i es provarà si funciona.



Figura 6.6. Mòdul amb el receptor NEO-6M i l'antena adquirits.

6.2. Transmissor de dades

És important saber quins són els serveis que s'ofereixen actualment per a la transmissió de dades a llargues distàncies. Quins són els més utilitzats en projectes similars i quin seria el més convenient.

Ja s'ha vist en apartats anteriors quins són els mètodes que existeixen per a enviar dades a llargues distàncies. Actualment es poden utilitzar tecnologies de segona, tercera i fins i tot quarta generació. Però, quina és la més adequada?

Quan es parla de segona generació ens referim a GPRS, una tecnologia adequada per a trucades telefòniques i SMS que permet velocitats de transferència de entre 20 i 70 kbps. Pel que fa al 3G parlem de UMTS i aquesta tecnologia més moderna permet velocitats promig de 1 Mbps

No ens decantarem per el 4G o LTE ja que bàsicament no es necessiten velocitats tan altes per a la transmissió de dades GPS. Pel que la decisió es troba entre el 2G i el 3G.

6.2.1. Requeriments del mòdul: 2G o 3G?

Vist com es troba el mercat i les empreses que gestionen els estàndards actuals, l'elecció entre 2G o 3G és força important i difícil pel que s'ha preferit estudiar-ho amb més profunditat.

- Mida.

Si més no la mida no és una característica molt important per al projecte, la mida entre un mòdul de transferència de dades és molt més gran que la d'un receptor GPS per exemple, i una mida capaç de ser portat a la butxaca seria l'adequada. Tot i així, els mòduls 2G i 3G no difereixen gaire amb la seva mida i seria poc raonable decantar la balança entre els dos per factors tant externs com aquest.

- Temps de vida.

Aquest factor és molt important considerar-lo. Aquests mòduls necessiten d'una targeta SIM per a funcionar, i com que el manteniment de dues xarxes que ja no són les que lideren el mercat costen diners per a mantenir, tot apunta que una de les dues desapareixerà en els pròxims anys. Quina? Bé sembla evident que el 2G per a ser més antic... I aquest pensament seria encertat a Estats Units, però no a Espanya, qui ha decidit apagar el 3G l'any 2020 i el 2G l'any 2025. Això és degut a que moltes companyies i components industrials utilitzen el 2G per a comunicar-se. A més a més, el mercat de mòduls i microprocessadors que utilitzin el 2G es troba molt més explotat que els de 2G, ara ho contrastarem. Pel que tot apunta que tot i que de menors prestacions, el 2G segueix essent una aposta segura a la península.

- Velocitat.

Unes desenes de Kbps a unes cent desenes de Kbps són el que diferencien el 2G del 3G. Si que és important la velocitat, però fins a quin punt? Les coordenades GPS que interessa enviar no ocupen més que un SMS de 50 paraules, i no estarem cada 10 ms refrescant les dades ja que per a enviar dades amb el servei de una companyia hi ha taxes que s'han de pagar, s'acabaria el crèdit de la targeta SIM amb molt poc temps i aquí entra un altre característica.

- Preu per quantitat de dades enviades.

Si ve són dos tecnologies diferents, les companyies no fan distinció entre la quota d'un usuari 2G amb la d'un usuari 3G o 4G. El preu ve determinat pel consum de dades (en contractes prepagament) pel que es pagarà el mateix ja utilitzem una o altra.

- Existències.

Mòduls 2G per a la utilització mitjançant microprocessadors n'hi ha molts i són fàcilment adquiribles per internet, i més ara que a alguns països com EEUU algunes companyies de telecomunicacions ja han apagat el seu servei 2G. Com que moltes indústries utilitzen aquest estàndard per a la comunicació M2M, aquest tipus de plaques electròniques s'han fet amb massa i no costa trobar-los a més a més a preus atractius.

Per altra banda, els mòduls 3G per a microprocessadors també existeixen però les existències que es poden trobar són molt més limitades i amb menys informació. Això fa que el producte tingui un preu molt més elevat i menys competitiu.

- Preu.

Parlem de preus que volten els 15 a 30 euros per a plaques senzilles que inclouen la funció de transmissió de dades mitjançant GPRS, és a dir 2G i de preus a partir dels 60 euros amb els productes estàndard a 100 i fins a 150 euros per a tecnologia 3G.

Ja es veu que la diferència de preu és important.

Amb aquestes especificacions d'ambdues tecnologies, s'ha decidit el següent per a la resolució del problema:

Es farà la compra d'un mòdul que utilitzi tecnologia de segona generació i es provarà i s'analitzarà com es comporta i després es pensarà amb un possible salt a tercera generació com a ampliació. A més a més amb una placa adequada 3G es podrien fins i tot millorar més coses del projecte.

6.2.2. Models de mercat 2G

Hi ha diferents versions de mòduls segona generació al mercat. Una de les marques més reconegudes és SIMCom que ha dissenyat al llarg dels anys diversos models que poden transmetre dades i veu mitjançant GPRS i GSM amb una targeta SIM d'una companyia telefònica.

Destaquen per el seu preu atractiu i per la seva simplicitat, i els dos models més venuts són el SIM800 i el SIM900 cadascun d'ells amb les seves versions. La diferència entre els dos models és pràcticament inexistent. El SIM800 és més recent i més econòmic a part de que incorpora un petit sistema Bluetooth opcional. Però pel que fa l'aspecte tècnic, els dos funcionen amb les mateixes condicions i amb els mateixos estàndards. La programació dels mòduls és també igual.

Molts dels distribuïdors d'aquest mòdul es troben a pàgines d'internet com ebay, amazon o aliexpress i venen els mòduls en plaques amb els components necessaris per a que el funcionament de la placa sigui còmode i fàcil, però també podem trobar plaques més elaborades com la de Cooking Hacks a preus menys competitius.

Un dels avantatges d'aquests mòduls és que poden operar arreu del món ja que incorporen en totes les quatre bandes GSM utilitzades globalment. (Technology: Quad-Band GSM/GPRS).



Figura 6.7. Mòdul SIM900 de venda online.

Cooking Hacks és la creació d'un petit grup de enginyers electrònics i de software que un dia es van proposar crear una companyia on l'electrònica pogués ser entesa per tothom. Aquesta ja no tant petita empresa online és l'encarregada de produir diferents mòduls amb targetes GPRS i UMTS per a la creació de projectes electrònics sense fils.



Figura 6.8. Mòdul SIM900 de cooking hacks. (Font: cooking hacks [61])

Aquests creadors de contingut han tingut una idea similar a la d'aquest projecte i s'han dedicat a crear mòduls adaptats per a la utilització de tecnologia sense fils amb GPS i altres sensors i actuadors com un altaveu i un micròfon. Remarcar que no et venen un producte funcional, et venen una placa adequada que tu li pots incorporar elements i programar per a que funcioni segons les teves especificacions.

El preu de la placa bàsica amb funcionalitat GPRS simplement es troba als 63€, mentre que les que incorporen un receptor GPS costen 159€.

Arduino també té la seva pròpia placa sense fils per a realitzar projectes amb el seu microprocessador. Es tracta d'una placa amb un mòdul de la companyia Quectel 2G amb antena inclosa que a la pàgina oficial es ven per a 72€ més impostos i enviament.



Figura 6.9. Mòdul 2G d'arduino. (Font: Arduino [63])

Hi ha altres plaques i mòduls clons i similars als descrit anteriorment, els quals semblen els més destacables i generalitzats.

6.2.3. Models de mercat 3G

Molts dels models 3G del mercat actual repeteixen distribuïdor com és el cas de adafruit o cooking hacks. Fins el moment, la gran majoria de transmissió M2M es feia amb estàndards 2G i el mercat en aquest sector s'havia fet tant competitiu fins al punt que un mòdul pot costar-te 8€ a grans tendes online. Però els mòduls 3G no són tant freqüents i això fa que molts distribuïdors de producció en grans quantitats no optin per a clonar aquest tipus de plaques. Potser amb l'apagat del 2G, aquestes empreses es posin a produir mòduls 3G, però no n'és el cas actualment.

Adafruit té una placa compatible amb UMTS, GSM i GPS per si no fos poc. Proporciona cobertura 2G per si no troba cap cobertura 3G disponible, i el GPS corre de la marca Qualcomm ja vista en anteriors apartats. Tot i així el GPS només té 16 canals i les característiques són força bàsiques.

Inclou pins per a la connexió d'altaveu i micròfon, connector jack per a auriculars i altres funcions extra. Un mòdul molt complet com ja ve fixant la companyia en tots els seus productes. L'únic inconvenient a destacar és que no diu res de les antenes i és possible que no vagin incloses.

El preu és de 106€.



Figura 6.10. Mòdul 3G GSM GPS de adafruit. (Font: Cetronic [64])

6.2.4. Elecció del mòdul

La característica que se'ns dubte influenciarà més en la tria d'aquest component, és el preu. Mentre que els dispositius creats per petites companyies i empreses amb renom costen entre 60€ i 200€, els mòduls de fabricació oriental produïts a gran escala costen al voltant dels 10€ a 20€. És una llàstima que no hi hagi models 3G d'aquest tipus. Potser en sortiran quan el 2G desaparegui completament. L'inconvenient d'aquests és, a part del temps d'enviament i la poca comunicació amb el venedor, que no és assegurat que el mòdul funcioni correctament, o que ho deixi de fer al poc temps.

Tot i així, si després el projecte es decideix ampliar i afegir una tecnologia de tercera generació, tampoc val la pena ara comprar un mòdul tant bo. L'elecció final resulta en un mòdul SIM900 de l'empresa SIMcom de fabricació xinesa. S'haurà de provar i veure com funciona.



Figura 6.11. Mòdul GPRS adquirit.

6.2.5. Targeta SIM

Ja se sap que aquests mòduls funcionen amb una targeta SIM igual que si es tractés d'un dispositiu mòbil per a connectar-se a la xarxa GPRS de la companyia telefònica.

El paper de la companyia és molt important perquè aquestes estan acostumades a un perfil de clients molt diferent al que s'està interessat en aquesta necessitat. Primer de tot falta detallar que en un principi només es requerirà connexió a internet pel que les trucades i enviament de SMS és un servei que no s'hauria de contractar si és possible per evitar factures innecessàries. Per altra banda, el mòdul 2G no obrirà ni transmetrà una alta quantitat de dades com podrien ser imatges o vídeos, tampoc és que pugui, pel que, novament per evitar una tarifa gaire inflada, s'hauria d'optar per una tarifa de poca quantitat de dades.

Movistar, una companyia que no té pensat eliminar les antenes 2G a Espanya ara mateix, ofereix una SIM amb 10 euros de saldo de contracte per 10 euros. Amb 10 euros, pots utilitzar la targeta quan vulguis sense haver de donar cap targeta de crèdit o compte bancari ja que tu la recarregues quan vols i no és l'empresa qui t'agafa diners cada mes. Després de comprar-la però em van dir que la tarifa de dades no estava activada i al trucar a la companyia els serveis que m'oferien eren molt dolents; Fins ara la targeta estava amb un pla de prepagament amb el qual segons el que consumia de trucades i SMS, pagava. Però amb les dades no et deixen fer això, i la tarifa més econòmica que inclou la utilització de dades són tres euros per setmana amb la possibilitat d'utilitzar una gran quantitat de MB. Com és evident, la quantitat de dades que utilitza el dispositiu per a connectar-se a

la xarxa és ínfima en comparació amb les que donen pensades per a connectar-se a internet diàriament amb un ús molt més elevat.

No té cap sentit seguir amb aquesta companyia (i sembla que qualsevol de les que controlin una alta quota de mercat) pagant aquest diners que no s'utilitzaran, per això s'ha decidit canviar de companyia i la escollida ha estat Symio perquè ofereix un gran ventall de possibilitats pel que fa al pagament de dades.

The screenshot shows the Symio app interface. At the top, the Symio logo is next to the tagline 'PAGA LO JUSTO' and social media icons. Below this is a navigation bar with 'TARIFAS' and 'MÓVIL'. The main section is titled 'CREA TU PROPIA TARIFA' and 'PREPAGO Y CONTRATO - IVA INCLUIDO - 4G DE SERIE'. It features two columns of tariff options: 'INTERNET' and 'LLAMADAS'. The 'INTERNET' column lists options from 100MB to 3GB, with the 100MB option selected and highlighted in orange. The 'LLAMADAS' column lists options from 30MIN to ILIMITADAS, with the 30MIN option selected and highlighted in orange. To the right, the account balance is shown as 'Saldo disponible 8,82€'. Below this, the 'Consumo Detallado' is shown as '0,19 €' for the current month. A detailed breakdown of consumption shows 'Llamadas' at 0h 00m 05s for 0,19 € and 'Datos' at 0,682 MB for 0 €. A note at the bottom states: 'El total de consumo no incluye descuentos ni promociones'.

Figura 6.12. Tarifes a escollir i pantalla client de Symio amb el consum i saldo disponible.

La tarifa contractada és la de 100 MB a un euro al mes, i com es veu a la foto de la dreta, on ja havia entrat i adquirit certes dades d'una pàgina web unes 5 vegades el consum de dades seguia a 0 €. Bé, si ho pensem millor, els 0.682 MB utilitzats serien uns 0.7 cèntims pel que, de manera molt aproximada i només per a fer-nos una idea, s'aproxima que cada cop que s'envien les nostres coordenades GPS a un servidor d'internet utilitzem 0.1 MB és a dir 0.1 cent, si es volgués enviar uns 30 cops al dia, es gastarien 3 cèntims al dia, aquesta planificació no gastaria l'euro contractat al mes deixant-nos a 90 cèntims per mes. Ja s'ha comentat que els càlculs són molt aproximats i més sense saber actualment quan gastaria l'enviament de les coordenades, però definitivament serà increïblement menor a la tarifa de Movistar.

Per acabar comentar que el que faria encarir aquesta tarifa seria la contractació de minuts per a poder parlar per telèfon. També es pot veure a la imatge que una sola trucada telefònica sense resposta, pel sol fet de trucar sense cap pla de trucades ha costat 20 cèntims d'un saldo d'on es recarrega també l'euro mensual per al consum de dades. I el pla més econòmic surt a 2 euros al mes que sumats al euro de dades passàriem a pagar 12 € de línia a l'any per 36 € triplicant el preu.

6.3. Alimentació

Hi ha diverses maneres d'alimentar tot el circuit, però el fet que sigui un dispositiu portàtil ja en limita les opcions a piles i bateries.

Les piles de 9 V són la opció més fàcil. El problema és que Arduino i el mòdul funcionen amb 5 V pel que no es podria alimentar directament, hauria de passar pel propi regulador de tensió, i aquest voltatge sobrant seria un malbaratament d'energia impensable.

Les piles recarregables tenen també problemes similars. Tot i que es poden reutilitzar, els mA/h són molt pocs pel que gairebé no tenen duració.

Les piles AA són una opció més viable. Proporcionen 1.5 V pel que caldria ajuntar quatre en sèrie per a que donés el 5 V d'alimentació. La vida útil de cadascuna d'elles és molt més extensa que les piles de 9 V, les primeres tenen uns 300mA/h i aquestes entre 2700 i 2900 mA/h. Bé, el problema de les piles és que es gasten i les recarregables normalment perden vida útil amb algunes càrregues i descàrregues. Si la gran majoria de dispositius sense fils utilitzen bateries, serà perquè és el mètode més eficient i, per tant, el que també s'utilitzarà en el projecte.

Al datasheet del fabricant del mòdul SIM (es pot veure al annex), s'ha comprovat que el mòdul pot tenir pics de fins a 2 A, pel que s'haurà de assegurar que la bateria pugui donar-los i no amorrar-se. Si no n'és el cas, s'haurà d'utilitzar un parell d'elles.

De bateries ni ha de moltes capacitats diferents. Les més usals tenen 1600 mA/h i aguanten molts cicles de càrrega i descàrrega.

Al final s'ha decidit comprar un model de 3000 mA/h per aconseguir exprimir al màxim l'autonomia del dispositiu, d'una marca que fabrica bateries de recanvi per a tablettes.



Figura 6.13. Bateria escollida. (Font: asturinformatica, ebay)

La càrrega de la bateria també s'ha de contemplar ja que farà falta material, per exemple un carregador.

Per diverses pàgines de venda, hi ha uns mòduls que tenen una entrada micro USB o mini USB com la que tenen els dispositius mòbils i una sortida per la connexió de bateries que permeten la seva càrrega. Es tracta de mòduls de càrrega lineal amb dos leds que indiquen si la bateria es troba carregada o descarregada. El preu és molt econòmic pel que per tots aquest motius s'ha decidit la seva compra.



Figura 6.14. Mòdul càrrega bateries. (Font: ebay)

6.4. Altres

En aquest apartat s'acabarà d'escollir o referenciar de manera clara la resta de material sobretot software que s'utilitzarà en el projecte de manera resumida ja que segurament ja s'ha detallat en apartats anteriors.

- Servidor d'internet. Si bé hi poden haver altres servidors gratuïts, com s'explica a l'estat de l'art els més utilitzats i de millor categoria són de pagament. Heroku ofereix un servei gratuït limitat amb el qual es basarà el projecte que tampoc és que requereixi de molts accessos o memòria. Per tant la decisió final es un servidor de Heroku.
- Software de desenvolupament de l'aplicació mòbil. El software escollit és Qt Creator ja que es porta un temps utilitzant i la resta de programes s'hauria de començar des de zero i sense ajuda ja que aquest és el programa utilitzat a les classes de la universitat.
- Per a realitzar l'alimentació del dispositiu es farà servir un fusible de 3A, un botó d'ancoratge i una placa de topes apart de tots els elements vists ja.

- Smartphone per a proves d'aplicació. No fa falta ni molt menys comprar-se un dispositiu per anar provant l'aplicació en el transcurs de la creació d'aquesta. Per a poder utilitzar un Smartphone amb aplicacions descarregades d'una IDE com Qt, fa falta posar el mòbil amb mode desenvolupador i activa la depuració USB. Així doncs es farà servir el propi dispositiu mòbil actual, un BQ Aquaris E5 HD.
- Ordinador per al desenvolupament del software. Igual que el mòbil, l'ordinador al que se li baixaran tots els programes per a poder programar l'aplicació mòbil és el que es disposa actualment i amb el que s'està redactant la memòria. Un ASUS X75V amb Windows 10 com a sistema operatiu.

7. Planificació de les tasques

El projecte pràctic es dividirà en tres parts en les que cadascuna d'elles tindrà les teves pròpies taques a completar. Per una banda la part Hardware o el Prototip, que ha de contemplar tots els elements per a que aquest funcioni, l'aplicació, que contempla les diferents finestres que seran necessàries (al propi apartat ja definirem cadascuna d'elles en detall), i la unió entre les dues plataformes, el servidor que ha d'incloure una base de dades i que el conjunt funcioni per enllaçar el microprocessador i l'aplicació.

Tot i que el projecte és desconegut i no existeix cap guia pel que no se sap encara en exactitud com s'anirà desenvolupant, s'ha fet un Diagrama de Gaant el qual reflexa les dates a les que aniria bé tenir enllestida cadascuna de les parts partint del dia 25/01/2017 que és quan s'està redactant aquest apartat.

Aquest diagrama es troba a la pàgina següent, i com es pot observar no és un diagrama lineal. Considerant que per a finalitzar el prototip fa falta que aquest envii les coordenades a un servidor correctament, no té cap sentit pensar que primer s'acabarà aquesta tasca i després es començarà el servidor.

Per altra banda, la creació d'un servidor no serà segurament una tasca fàcil pel que se li ha donat de marge quasi un mes per a tenir-ho enllestit. Un cop acabat es podrà tornar a dedicar amb el prototip per a acabar-lo.

Pel que fa a l'aplicació, és molt probable que tampoc sigui lineal. El problema de no avançar cada part a la vegada és que si sorgeix algun dubte i no el pots solucionar fins que no puguis parlar amb un professor, et responguin en un correu, és temps perdut que no es pot perdre tenint en compte que es disposa de tres mesos per a tenir acabat un treball molt elaborat. Si s'avança per totes les àrees, el rendiment personal és major també.

Així doncs, essent el màxim de realistes possibles, la planificació de totes les tasques del treball final de grau queda tal i com reflexa el diagrama de Gaant a continuació.

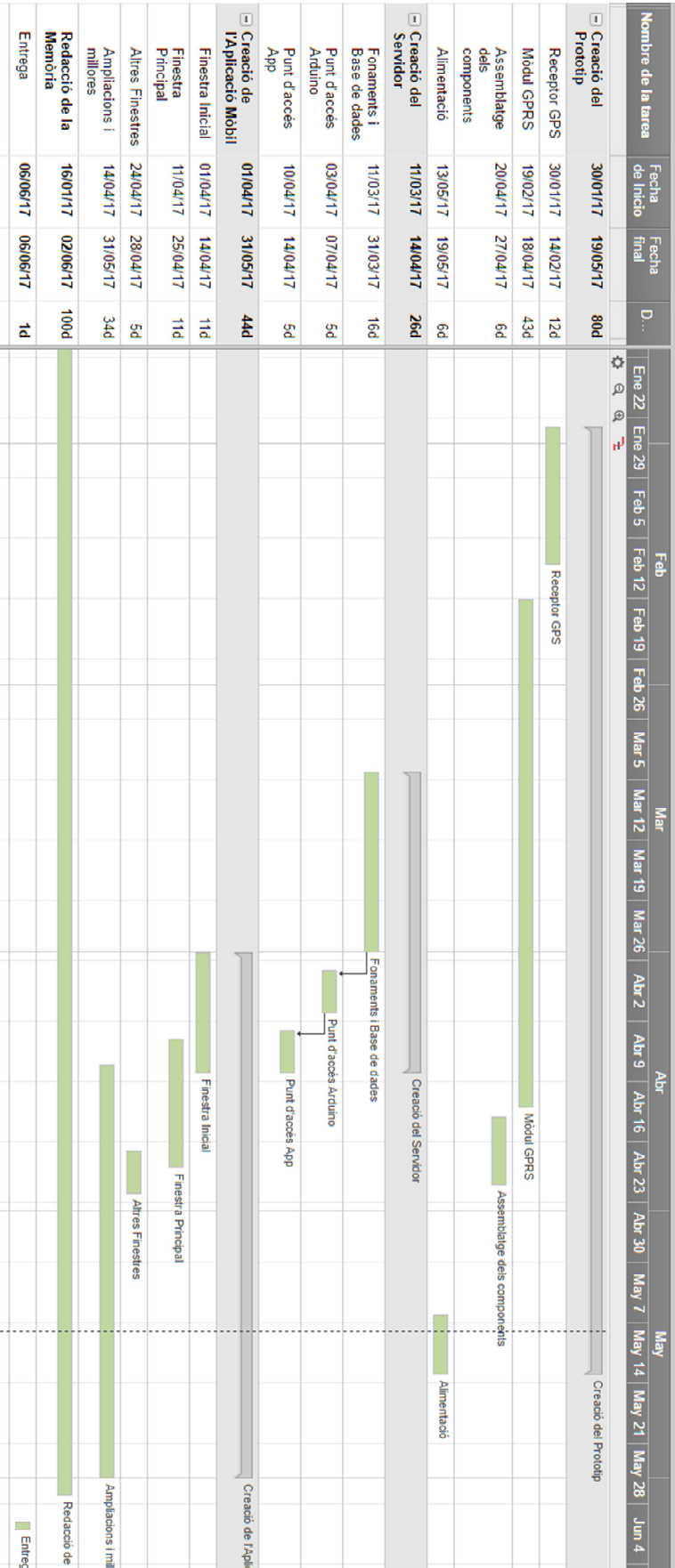


Figura 7.1. Diagrama de Gaant.

8. Posada en marxa

En aquest apartat es provaran tots els components electrònics per a veure el seu funcionament. S'analitzarà el seu disseny i se'ls sotmetran a diferents proves i tests inicials per a després poder programar de manera adequada.

8.1. Receptor GPS

Abans de començar a testejar el receptor GPS hi ha alguna informació del procés que convé explicar.

Com ja s'ha vist a fonaments, els receptors GPS reben senyals de diferents satèl·lits per a calcular la posició a la qual es troben, però aquesta informació s'ha de saber interpretar. El que els satèl·lits envien es coneix amb el nom de NMEA. Aquest acrònim ve de National Marine Electronics Association i es tracta d'una associació creada al 1957 per un grup d'electrònics per a crear una millor comunicació entre els fabricants i els comerciants. Avui dia, NMEA es un estàndard recolzat per a tots els fabricants de receptors GPS, instruments de navegació, sondes, etc. Amb el qual es comunica el receptor. NMEA ve a ser el que és el codi ASCII en caràcters informàtics digitals, però amb el format de dades per a la comunicació de dispositius GPS.

Per tant, entendre com funciona la comunicació amb NMEA serà clau per al bon desenvolupament del software que el tradueixi. Per comprendre l'estructura dels missatges NMEA, examinarem un dels missatges rebuts per a un receptor GPS:

\$GPGGA,221510.00,4124.8963,N,08151.6838,

*W,4,11,1.1,480.143,M,29.200,M,0.10,0000*40*

- Tots els missatges NMEA comencen amb el caràcter \$, i cada camp de dades és separat per una coma.
- *GPGGA* indica de quin satèl·lit prové, en aquest cas d'un americà Global Positioning System Fix Data (GL denotaria GLONASS).
- *221510.00* és la marca de temps en hores, minuts i segons sense contar els dos últims zeros. Quedaria doncs les 22h 15min i 10s.
- *4124.8963* és la latitud i la lletra *N* indica Nord. Si ho traduïm quedaria així 41d 24.8963' N. Pot ser d'hemisferi Nord o de Sud.
- *08151.6838* és la longitud i la lletra *W* indica Oest. Si ho traduïm quedaria així: 81d 51.6838' W igual que l'altitud. Amb aquestes dues coordenades ja podríem situar el nostre receptor GPS a un mapa.

- 4 denota l'Indicador de Qualitat de la següent manera descrit per l'estàndard NMEA:
 - o Nombre 0: Indica que no s'ha trobat una posició fixe disponible.
 - o Nombre 1: Indica que s'ha trobat una posició fixe disponible SPS. (Línia civil L1, és a dir, el servei estàndard).
 - o Nombre 2: Indica que a més de rebre una senyal fixe, el receptor es troba en mode diferencial.
 - o Nombre 3: Indica una posició fixe PPS. (Línia militar L2, és tracta del servei més precís per a fins no civils).
 - o Nombre 4: El receptor es troba en mode real time kinematic amb punts fixes.
 - o Nombre 5: El receptor es troba en mode RKT però amb punts flotants.
 - Seguint amb la combinació d'exemple, 11 denota el nombre de satèl·lits utilitzats en la coordenada.
 - 1.1 denota la HDOP o dilució horitzontal de la precisió i contempla valors entre el 00.0 al 99.9.
 - 480.143 denota l'altitud de l'antena a nivell del mar.
 - M representa unitats d'altura (per exemple. Metres o peus)
 - 29.200 denota la separació del geoide (restar aquest valor de l'altitud de l'antena per arribar a l'altura per sobre del el·lipsoide HAE).
 - 1.0 denota l'edat de la correcció en mode diferencial i és nul si no s'utilitza en aquest mode.
 - 0000 denota la identificació de l'estació de correcció també en mode diferencial.
 - *0E denota la suma de comprovació o checksum comunament utilitzats en telecomunicacions per a comprovar que el missatge s'hagi enviat correctament.
- Per a obtenir aquest valor s'ha de fer una OR exclusiva de tots els valors compresos entre el signe del dollar \$ i l'asterisc *.

El GPGLL \$ és un missatge bàsic del GPS NMEA. Hi ha missatges NMEA alternatius i complementaris que proporcionen informació similar o addicional [65].

Explicat el funcionament de l'estàndard NMEA podem posar-nos amb el mòdul GPS.

8.1.1. Hardware del mòdul GPS

Tots aquests mòduls que es venen a preus assequibles porten un circuit imprès amb diferents resistors, LED una memòria EEPROM per a guardar les dades de configuració en l'apagat i externament una antena per a captar les senyals GPS.

Encara que la companyia del receptor no proporciona informació sobre l'esquema òptim per a la implementació del mòdul, en dona una mica d'informació, i per internet es poden trobar un o dos esquemes de connexió.

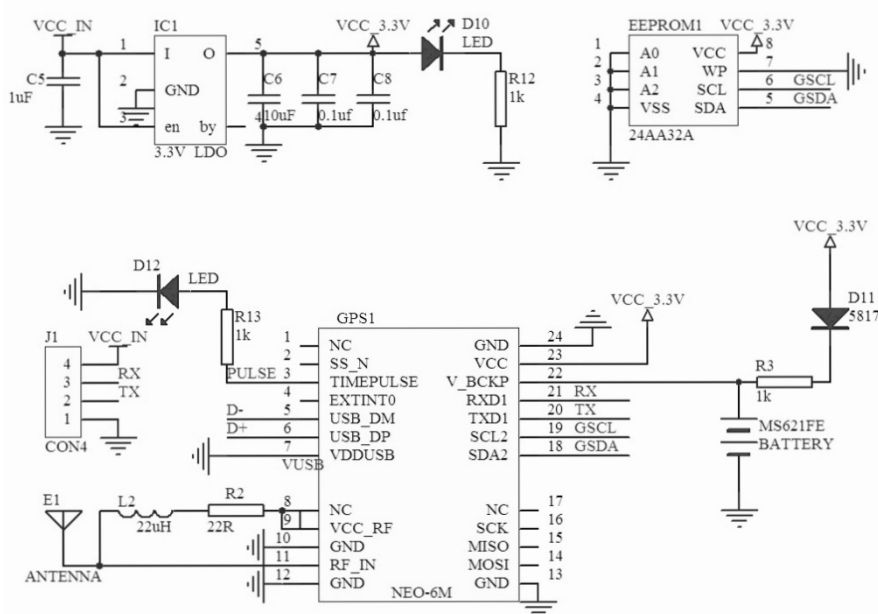


Figura 8.1. Esquema elèctric del mòdul NEO-6M. (Font: cnx-software [66])

És fàcil entendre el funcionament del circuit i com connectar els components amb les indicacions del venedor; Se sap que el mòdul GPS funciona a 3.3 V i que normalment es connecta als 5 V d'alimentació. El primer esquema correspon a un regulador de tensió com podria ser perfectament el LM7833. En aquest cas no s'indica el nom del component. Els condensadors que porta l'esquema són valors típics dels reguladors de tensió. C5, abans del regulador, filtra la tensió d'entrada de possibles transitoris i pics indesitjables, mentre que C6, que es troba a la sortida, disminueix la tensió d'arissat de sortida, alhora que evita oscil·lacions. El valor del condensador de sortida depèn en part de la intensitat consumida en el circuit posterior, m'imagino que els dos altres petits condensadors faran funcionar a la pràctica una mica millor el circuit tot i que opino que es podrien eliminar. El LED que hi ha a la sortida, es posa molts cops per a saber quan el mòdul es troba alimentat (és a dir, si funciona tot el circuit d'alimentació) però aquest és un gest de consum que es pot perfectament evitar juntament amb la resistència.

A la part dreta del esquema es connecta una memòria EEPROM que utilitza el mòdul per a guardar la configuració de paràmetres establerts. La connexió és molt senzilla, funciona a 3.3 V igual que el mòdul i es connecta a aquest amb els pins GSCL i GSDA.

Per últim tenim el connexionat del mòdul. El pin 22 que s'anomena V_BCKP; Segons el datasheet del mòdul NEO-6M de UBLOX, s'utilitza quan el mòdul no es troba alimentat per a guardar els valors de configuració a la RAM [67]. Per a tenir alimentació utilitza una petita bateria de liti recarregable connectada a 3.3 V juntament amb un díode per a evitar que la corrent circuli en direcció al regulador de tensió quan funcioni la bateria. La resistència es posa per a limitar la corrent de carga de la bateria.

El pin 3 del mòdul declarat com a TIMEPULSE es tracta com un pin de sortida que el mòdul té per a donar diferents pols positius de la manera que es vulgui i estigui configurada. Els pols poden ser de 0.25 Hz fins a 10 MHz segons el datasheet i si se li posa un LED com al exemple (amb la seva pertinent resistència), quan el mòdul tingui fixats els satèl·lits per a donar la posició, s'il·luminarà intermitentment.

Per últim tenim el connexionat de l'antena amb una bobina de 22 uH i una resistència de 22 ohms. Pel que s'ha anat veient, molts models funcionen amb aquests dos components passius.

8.1.2. Comprovació del correcte funcionament del receptor GPS

Abans de començar a dissenyar el codi per a que el mòdul funcioni correctament, s'haurà de comprovar que aquest funcioni. Ublox dissenya els seus mòduls de manera que un cop es trobin alimentats, aquests enviïn per el port sèrie les dades que rebin dels senyals dels satèl·lits (si n'ha trobat) pel que s'ha considerat en simplement alimentar el mòdul amb 5 V i comprovar si envia dades.

És important saber que aquests mòduls funcionen amb 3V3 però com que incorpora un regulador de tensió, no fa falta rebaixar el voltatge per alimentar-lo i es pot fer directament amb 5 V.

Teòricament, si el mòdul funciona correctament, un cop l'alimenti es posarà a buscar tants satèl·lits com pugui trobar des de la posició a la que es troba, i si en troba tres o més serà capaç d'enviar per el seu port de comunicacions el NMEA trobat. Quants més satèl·lits fixats pel mòdul, aquest podrà oferir més qualitat en la precisió de la mesura.

Amb la placa d'Arduino UNO, s'ha alimentat el mòdul i connectat directament al port de comunicacions serial del microprocessador els propis canals de comunicació del mòdul, és a dir, Tx del mòdul GPS a Rx del microprocessador i viceversa tot i que aquest últim no era necessari i sense programar cap línia de codi, (recordem que un cop alimentat el mòdul envia NMEA sense la necessitat de rebre cap ordre) s'ha compilat el programa per a veure que succeïa al monitor sèrie d'Arduino.

Aquest n'ha estat el resultat al cap d'un minut d'espera:

```

sketch_jan20a $
void setup() {
}
void loop() {
}
Subido
El Sketch usa 716 bytes (2
Las variables Globales usa
3
$GPGSV,3,2,09,24,39,128,16,25,65,307,16,29,43,199,21,31,16,309,16*70
$GPGSV,3,3,09,32,43,271,20*40
$GPGLL,4120.69439,N,00141.53173,E,201109.00,A,A*63
$GPRMC,201110.00,A,4120.69399,N,00141.53105,E,2.018,,200117,,A*70
$GPVTG,,T,,M,2.018,N,3.737,K,A*28
$GPGGA,201110.00,4120.69399,N,00141.53105,E,1,08,1.09,216.0,M,49.5,M,,*5F
$GPGSA,A,3,12,25,24,32,29,14,02,31,,,,,2.20,1.09,1.91*0E
$GPGSV,3,1,09,02,27,086,20,06,15,049,,12,55,047,16,14,32,303,12*76
$GPGSV,3,2,09,24,39,128,15,25,65,307,19,29,43,199,22,31,16,309,17*7E
$GPGSV,3,3,09,32,43,271,20*40
$GPGLL,4120.69399,N,00141.53105,E,201110.00,A,A*67
$GPRMC,201111.00,A,4120.69413,N,00141.53092,E,1.336,,200117,,A*77
$GPVTG,,T,,M,1.336,N,2.475,K,A*20
$GPGGA,201111.00,4120.69413,N,00141.53092,E,1,08,1.09,215.8,M,49.5,M,,*5F
$GPGSA,A,3,12,25,24,32,29,14,02,31,,,,,2.20,1.09,1.91*0E
$GPGSV,3,1,09,02,27,086,20,06,15,049,,12,55,047,16,14,32,303,14*70
$GPGSV,3,2,09,24,39,128,14,25,65,307,19,29,43,199,23,31,16,309,18*71
$GPGSV,3,3,09,32,43,271,22*42
$GPGLL,4120.69413,N,00141.53092,E,201111.00,A,A*6C
$GPRMC,201112.00,A,4120.69417,N,00141.53127,E,0.592,,200117,,A*76
$GPVTG,,T,,M,0.592,N,1.096,K,A*23
$GPGGA,201112.00,4120.69417,N,00141.53127,E,1,08,1.09,215.2,M,49.5,M,,*5D
$GPGSA,A,3,12,25,24,32,29,14,02,31,,,,,2.20,1.09,1.92*0D
$GPGSV,3,1,09,02,27,086,20,06,15,049,,12,55,047,16,14,32,303,15*71

```

Figura 8.2. Monitor Sèrie d'Arduino amb la connexió del mòdul GPS al port de comunicacions.

Tot i que el minut d'espera abans de rebre dades s'ha fet llarg, és lògic pensar amb un temps similar tenint en compte el TTFF del mòdul i que es trobava dins una casa.

Efectivament el mòdul funciona i rep dades dels satèl·lits. No podem assegurar encara que funcioni correctament, però no enviaria res si no fos capaç de rebre senyal dels satèl·lits i/o d'enviar-la.

Anem ara a analitzar la informació obtinguda en NMEA.

El primer \$GPGGA es troba a la fila número sis de la foto, tot i que hi ha molts altres NMEA, no tots tenen la mateixa informació, per exemple \$GPGSA, que és un NMEA dels que més es repeteix, conté dades generals de recepció del satèl·lit o \$GPGSV que conté dades detallades del satèl·lit. Els que interessa doncs són els primers, i més concretament la latitud i la longitud, les quals es troben abans de la lletra N i de la lletra E que signifiquen Nord i Est.

La diferència entre les lletres que segueixen els nombre de la latitud i la longitud tenen una explicació. Si la lletra de la latitud és N, el valor del nombre serà positiu, si en canvi és una S, el valor del nombre serà negatiu, i amb la E i la O de la longitud passa el mateix essent la E la positiva. Això és perquè Google Maps i tota la resta de programes de mapes tenen com a referència de Nord a Sud i d'Est a Oest mentre que els receptors t'ho poden indicar del revés [68]

Amb això explicat, les coordenades quedarien així:

Latitud: 4120.69339 El qual s'interpreta com 41° i 20.69339'

Longitud: 00141.53105 El qual s'interpreta com 001° i 41.53105'

En el següent codi \$GPGGA la diferència entre la latitud i la longitud és petita, però existent. Concretament la latitud passa a valdre 41° i 20.69413' i la longitud 001° i 41.53092'.

Per a provar definitivament que el mòdul funciona correctament es prova a posar aquestes coordenades al Google Maps a veure on es troba aquesta geolocalització.



Figura 8.3. Imatge de Google Maps amb les coordenades rebudes pel receptor GPS.

Doncs ha donat en el clau! I per a observar amb més detall la precisió de les coordenades rebudes s'ha fet un zoom de la posició en el mapa.



Figura 8.4. Imatge de Google Maps amb les coordenades rebudes pel receptor GPS ampliat.

Es pot veure que el punt es troba exactament entre dues cases i hauria d'estar més o menys al mig de la casa de més a l'esquerra. Això és un error aproximadament de mig metre per mig metre. La resolució és molt encertada i s'ha fet més proves que encara donen una millor resolució amb un marge d'error inapreciable.

Abans de començar amb la programació del mòdul, hi ha hagut un altre procediment que s'ha dut a terme per a veure com d'eficient i complet era el mòdul GPS. El fabricant oficial del mòdul, té un programa de PC de lliure accés anomenat U-CENTER on si se li proporciona via FTDI un mòdul, el programa representa diverses funcions i captures de la localització.

Amb un connector FTDI i un cable mini USB a USB, els que es mostren a la imatge de a continuació, s'ha connectat al ordinador el mòdul i s'ha pogut comprovar tot el funcionament i enregistrament de dades. També es deixen dues imatges per a veure moltes de les opcions que dona el programa.

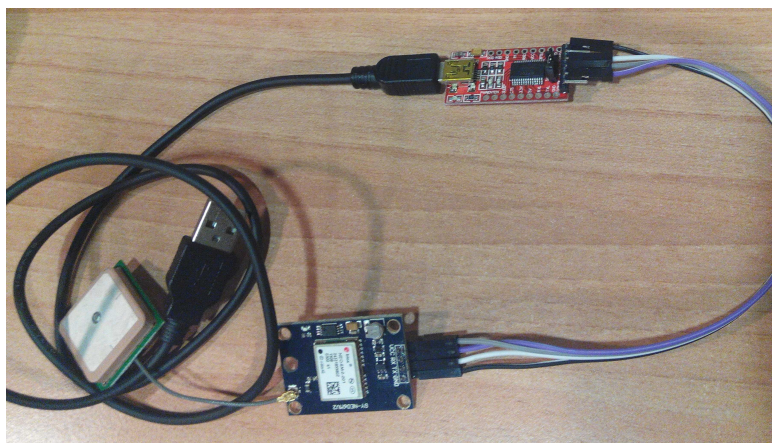


Figura 8.5. Imatge del mòdul FTDI juntament amb el mòdul i un connector mini USB a USB.

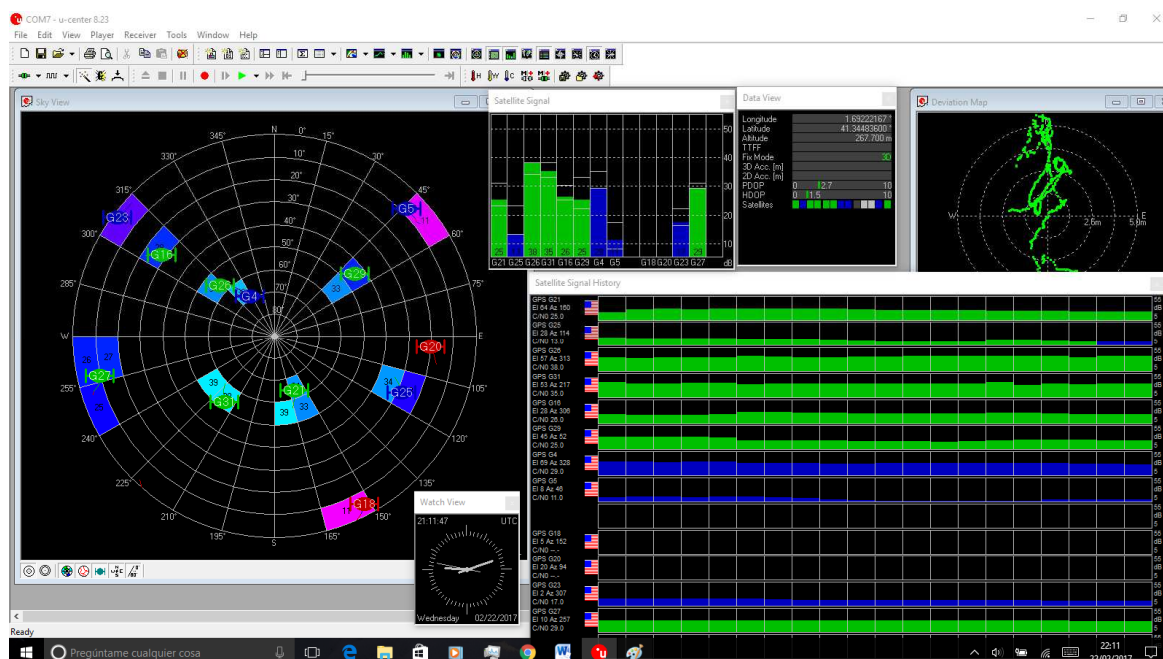


Figura 8.6. Instantània del programa U-CENTER amb els satèl·lits.

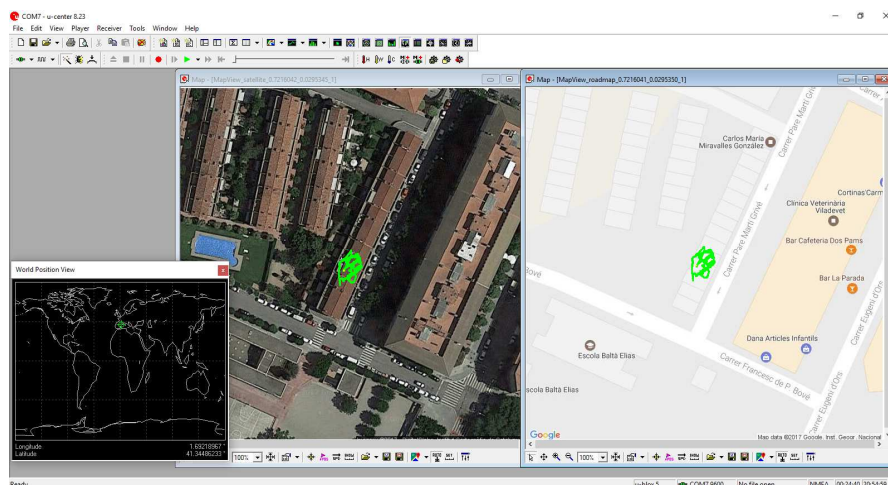


Figura 8.7. Instantània del programa U-CENTER amb la posició.

La primera imatge del programa, la pantalla a l'esquerra mostra la posició de tots els satèl·lits que reconeix el mòdul a temps real i la seva qualitat de transferència de dades. El cercle que els rodeja significa l'horitzó i la posició dels satèl·lits està en graus. La qualitat de cada satèl·lit és més o menys relativa: Primer fa falta que no hi hagi cap element prou voluminós i opac com per a que no deixi transmetre correctament els NMEA, també importa l'angle i la distància a la que es trobi, ja que qualsevol satèl·lit serà incapaç de transmetre informació a la cara oposada de la Terra per qüestions físiques simplement. D'aquí doncs, els satèl·lits que es trobin just a sobre del receptor són els que tenen més possibilitats de transmetre NMEA correctament. El programa distingeix entre qualitat bona, feble i dolenta amb els colors verd blau i vermell respectivament. Les altres pantalles mostren també els satèl·lits que troba en temps real i la seva qualitat d'altres maneres, i la imatge de més a la dreta mostra la precisió del mòdul en funció dels NMEA i ho representa en un cercle que va movent el seu punt mig en funció de la mitja gràfica de tots els punts enregistrats. Val a dir que per a enregistrar tots aquests punts, el mòdul ha estat prenent dades entre 5 i 10 minuts.

La segona imatge ensenya la localització en un mapa d'una manera molt semblant a com es té pensat fer en aquest treball. La imatge petita es veu la posició en el globus, i una imatge de Google maps i Google Earth amb la posició enregistrada durant tot el temps que ha estat el mòdul operatiu.

El mòdul ja s'ha pogut veure i comprovar que funciona perfectament pel que es podrà començar la seva programació per la correcte interpretació dels NMEA.

8.1.3. Disseny inicial del software GPS

La comunitat d'Arduino s'ha fet en els últims anys increïblement extensa i això ha fet que molt del software que s'utilitza, a part de Open-Source, sigui molt complet.

En el cas dels receptors GPS i a la interpretació de NMEA hi ha diferents llibreries que faciliten la feina en gran mesura. Una d'aquestes llibreries s'anomena tinyGPS i optimitza molt bé tot el codi que compila amb les seves instruccions.

A la pàgina oficial de la llibreria, s'explica com programar utilitzant aquesta eina i que han tret una millor versió, tinyGPS++ amb el que hi ha instruccions més simples tot i que a canvi de perdre una mica de memòria.

Al annex es pot trobar les diferents línies de codi per a extraure del codi en NMEA informació desitjada [69].

Per a fer funcionar però el programa s'ha de utilitzar més línies de codi com són la de "encode()" que passarem a veure amb més detall amb el codi que s'ha programat per a fer funcionar el receptor correctament.

El codi comença amb les llibreries que s'utilitzen: `TinyGPS++.h` i `SoftwareSerial.h`. La primera s'utilitza per a la interpretació de NMEA amb línies de codi més senzilles, però i la de `SoftwareSerial`?

Arduino té un canal de comunicacions sèrie (Rx i Tx) que utilitza per a comunicar-se de manera TTL amb qualsevol altre port sèrie de la mateixa tecnologia. (No confondre amb RS232). Però podem estalviar ocupar-lo creant-ne altres de virtuals. Realment la feina del port de comunicacions no és més que la d'enviar senyals binàries (0 V per un estat baix i 5 V per un d'alt) pel canal transmissor Tx, i la de rebre'ls per el canal receptor Rx. Aquesta operació la pot fer qualsevol altre pin del microcontrolador programat correctament, el qual anomenarem com a port virtual. Amb la llibreria `SoftwareSerial` la creació d'un port virtual al que després podrem enviar i rebre informació es fa amb una sola línia de codi:

```
SoftwareSerial Nom_del_port(Rx, Tx);
```

On s'hi escriurà el nom del port que li vulguem donar, i al lloc de Rx el numero que correspon al Pin de recepció de dades i al lloc de Tx el nombre del Pin de transmissió del microcontrolador.

També s'ha de declarar el GPS per a després poder donar-li instruccions, això és el que fa la següent línia de codi del programa.

```
TinyGPSPlus Nom_del_GPS;
```

A continuació es posa ja a programar la funció `setup()` on es declaren simplement els ports de comunicació virtual del GPS i del monitor sèrie d'Arduino per a que vagin a una velocitat de refresc de 9600 bauds.

Per a tenir-ho tot més ordenat, dins la funció `loop()` per ara només es cridarà la funció que realitza les instruccions de funcionament del receptor GPS. D'aquesta manera, el codi acabarà essent la crida de diverses funcions totes elles a la funció principal del loop.

Tal i com requereix la pàgina oficial de la llibreria del GPS, per a poder interpretar correctament el NMEA i mostrar-lo pel monitor sèrie, el primer que s'ha de fer és comprovar si hi ha dades provinents del port on es troba connectat el receptor GPS, en aquest cas, pins 10 i 11. Aquesta instrucció es pot fer amb la següent línia de codi:

```
while(Nom_del_port.available())
{
  gps.encode(portGPS.read());
}
```

Que demana que mentre el port GPS rebi senyal pel canal Rx (i conseqüentment voldrà dir que el receptor GPS ha trobat una senyal fixe de tres o més satèl·lits i l'esta emetent pel seu corresponent port Tx), es mantingui llegint, és a dir guardant la informació. Aquesta no és més que tot el codi NMEA. Un cop l'ha guardat, ja podrà sortir del bucle.

Un cop acabada la instrucció `while()`, si les dades que han estat rebudes canvien respecte les que ha llegit anteriorment (en una de les anteriors passades que ha fet; Així s'evita que estigui constantment mostrant el mateix NMEA una i altra vegada), mostri per el monitor sèrie d'Arduino les dades que desitgem. Tot això es fa amb les últimes línies de codi següents:

```
if(gps.location.isUpdated())
{
  Serial.println(gps.satellites.value());
  Serial.println(gps.location.lat(), 6);
  Serial.println(gps.location.lng(), 6);
}
```

En el nostre cas voldrem saber quants satèl·lits es troben fixats pel receptor, el qual s'indica amb la primera instrucció, la latitud i la longitud en coordenades en graus, el qual s'indica amb les darreres línies de codi.

Un cop depurat i compilat, el resultat és el que es mostra en la següent imatge presa del monitor sèrie del programa.

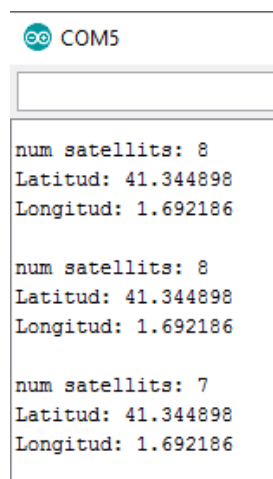


Figura 8.8. Imatge del monitor sèrie d'Arduino amb el codi anterior.

Ja tenim el que serà la base del software que utilitzarà el nostre equip en la presa de dades de localització GPS [70].

8.2. Mòdul SIM900

Aquest mòdul GPRS s'anomena exactament "GPRS SIMCOM SIM900 Quad band GSM shield" i es tracta d'una targeta ultra compacta de comunicació sense fils. A fonaments s'ha vist que la tecnologia GPRS és de segona generació (de fet 2.5G) i això és una avantatge i un inconvenient a la vegada. La part positiva és que d'aquests mòduls se'n porta un bon temps utilitzant, pel que els mòduls estan molt ben dissenyats i perfeccionats, a més es troba força material didàctic per la web, i la negativa és que la tecnologia es troba desfasada. Tot i així és una bona placa que admet la possibilitat d'enviar missatges SMS, rebre i fer trucades amb el seus pertinents altaveu i micròfon i fins i tot connectar-se i navegar per internet.

La targeta GPRS està configurada i controlada via UART utilitzant comandes AT, i això és del que es parlarà abans de fer funcionar el mòdul.

UART són les sigles de "Universal Asynchronous Receiver-Transmitter" que signifiquen receptor-transmissor asíncron universal. Es tracta del dispositiu comunament integrat a la placa base o al microprocessador que controla els ports i dispositius sèrie. La seva funció és la de transmetre els bits de cada byte d'informació que es vol enviar dels ports o dispositius al microprocessador i viceversa. En la recepció, un altre UART reconverteix els bits en bytes per a la seva interpretació (d'aquí el seu nom Receiver-Transmitter).

Els UARTs són comunament utilitzats juntament amb estàndards de comunicació com el EIA, el RS-232, RS-422, etc.

Quan parlem de comandes AT, ens referim a un llenguatge de programació desenvolupat per la companyia Hayes Communications utilitzat pràcticament per a configurar i parametritzar mòdems.

El nom de AT ve de que totes les ordres comencen per aquestes sigles que signifiquen Attention. Moltes de les ordres AT estàndard no tindria cap sentit utilitzar-les però sí que val la pena fer una llista de les ordres que es poden utilitzar en el projecte. La llista es pot trobar al Annex [71].

8.2.1. Hardware del mòdul SIM900

L'esquema electrònic del mòdul es pot trobar al llistat de plànols de l'Annex. Ara detallarem el seu funcionament, o almenys, el funcionament de les parts que es fan servir ja que el mòdul per exemple, contempla dos ports de transmissió i recepció de dades però es fan servir en el projecte dos pins per a fer-ho per software. O els connectors del micròfon i altaveu no s'utilitzen.

L'esquema contempla una part de potència, a l'entrada, un fusible de 3A, (el mòdul és capaç de donar pics de fins a 2A) i un díode en sèrie per evitar corrents inverses. Es connecta a l'entrada d'un regulador de tensió MIC29302. Aquest component suporta altes intensitats (fins a 3 A) amb pèrdues de tensió molt baixes (d'uns 300 mV fins a 370 mV en plena carga), necessari per a poder aguantar les puntes de corrent del mòdul. A la sortida s'obté la tensió d'alimentació del mòdul (3.3 V). El condensador a l'entrada serveix per a la estabilització, ometre'l acabaria amb el regulador en auto oscil·lació. Els de sortida són per a estabilitat a la sortida i pels transitoris d'altres i baixes freqüències.

Es pot trobar informació referent al component en el datasheet de la bibliografia [72].

Pel que fa al port de comunicacions sèrie, ens interessa el SRXD i el STXD que fan referència a Software RX i TX. Els podem veure al J11 com estan units al HRXD i HTXD respectivament. Aquests Pins venen del microprocessador.

Pel que fa a l'encesa del mòdul, es fa pel pin D9 per software i pel botó SW-PB que al ser premut tanca el circuit de POWKEY per la base i satura el transistor. Si en canvi ho és per software, simplement D9 va connectat a la base per a saturar el transistor quan arribi un pols alt.

En el centre hi ha el circuit per a la targeta SIM. SIMVCC, SIMRST i SIMCLK són entrades del circuit i sortides del microprocessador CLK es refereix al rellotge per a la velocitat d'enviament de dades, RST a Reset i VCC a alimentació. SIMDATA és el port d'entrada i sortida i envia les dades que rep la targeta SIM i el microprocessador envia dades com el codi Pin a ella. La matriu de díodes Zener és, simplement per a reduir espai i protegir fins a cinc línies (en aquest cas quatre) de transitoris de voltatge induïts per elles [73].

Per últim hi ha un circuit amb un LED a la part inferior del esquema que encén el LED si satura el transistor. Aquest saturarà quan el Pin SYNC sigui alt. Aquest circuit està pensat per a fer que el LED pampallugui en funció de SYNC que es troba connectat al microprocessador a la pota netLED o LED de la xarxa. Valdrà un segon quan no estigui connectat a la xarxa i quatre quan sí ho estigui.

8.2.2. Comprovació i primeres proves del mòdul SIM900

Aquesta placa tant completa, té diferents aplicacions i funcionalitats pel que per a comprovar que el mòdul no és defectuós i per a realitzar una primera prova d'aquest, es dissenyaran en concret quatre aplicacions, cadascun utilitzarà diferents comandes AT.

8.2.2.1. Primera aplicació. Connexió i prova de diferents comandes AT

Abans de connectar la placa a Arduino, es pot connectar al ordinador mitjançant un FTDI. S'alimenta el mòdul correctament i utilitzant el programa Tera Term, podrem comunicar-nos amb el mòdul per a enviar-li diferents comandes AT i veure que funciona.

Per començar, s'alimenta el mòdul amb una pila industrial duracell de 9 V i posant el botó d'alimentació del mòdul correctament, ja veiem que aquest s'encén. Si polsem el botó per encendre la connexió GPRS durant dos segons es pot veure que un LED comença a pampallugar cada segon. Això vol dir que s'està intentant connectar, en el moment en què passi a alternar cada quatre segons, voldrà dir que ho ha aconseguit, però tal i com està ara, no ho aconseguirà mai.

És moment de connectar el FTDI tenint cura de connectar GND, RX i TX al lloc que li correspon basant-nos amb el datasheet del fabricant i amb la placa d'Arduino UNO ja que a la placa no s'especifiquen els pins. Haurem unit doncs les masses del USB del ordinador amb la del mòdul i preparat la connexió sèrie d'aquest.

S'entra al programa TeraTerm i sortirà aquesta pantalla.

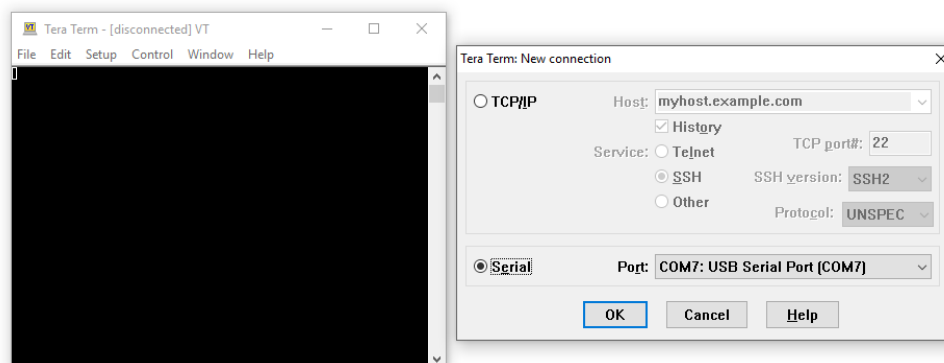


Figura 8.9. Imatge del programa Tera Term al inicialitzar.

Posem la opció Serial i premem OK.

Dins la pantalla negra del programa hem d'entrar a Setup i configurar primer el Baudrate al que tingui per defecte el mòdul, en el nostre cas, 115200 en la opció de Serial Port.

La primera comanda que s'haurà d'enviar serà la que permeti al mòdul connectar-se a la xarxa. Si es pensa, en qualsevol dispositiu mòbil el primer que es demana quan s'encén i és justament per a connectar-se a la xarxa de telecomunicacions, és el número PIN. La primera comanda que s'introduirà doncs serà la següent:

AT+CPIN=9812

Amb la qual es definirà el número PIN de la SIM integrada a la placa. S'haurà d'esperar un moment i ja es veu que el LED s'alterna cada 4 segons i no 1 com feia fins ara. El mòdul s'ha connectat correctament a la xarxa 2G i ja està llest per a ser utilitzat.

Si bé es podrien utilitzar comandes per a realitzar trucades, enviar SMS o connectar-se a internet, això es farà a les següents proves ja amb Arduino que és tal i com funcionarà al final i en aquesta provar amb diferents comandes com:

AT

AT+CGMI

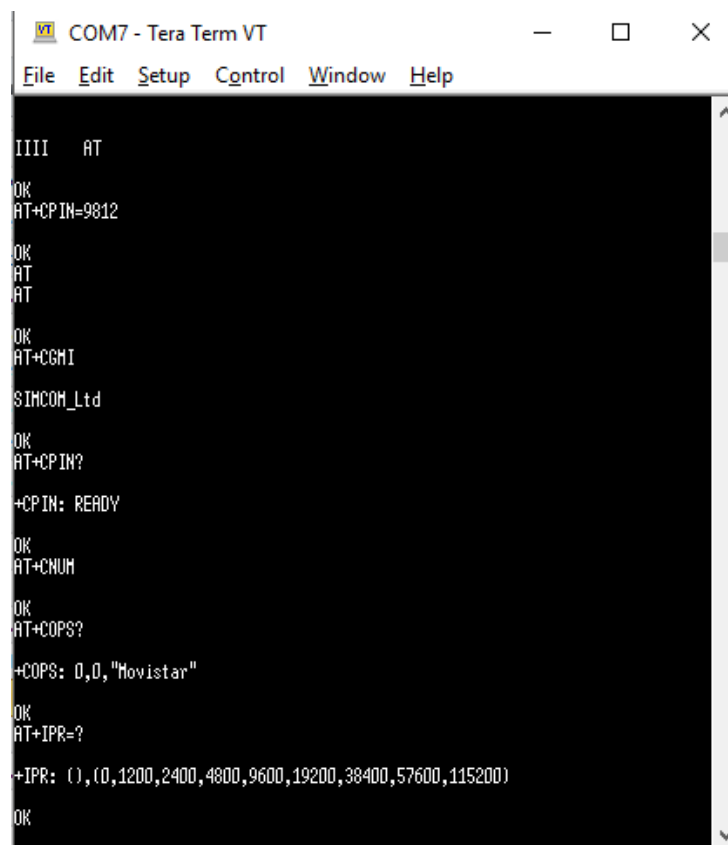
AT+CPIN?

AT+COPS?

AT+IPR=?

La primera i més bàsica comanda pregunta per l'estat del mòdul, i si tot funciona correctament, el mòdul hauria de retornar un OK. La segona indica qui és el fabricant del mòdul, en el nostre cas SIMCOM i això és el que hauria de mostrar. Comprova l'estat de la targeta SIM actual. AT+CPIN? comprova l'estat de la targeta SIM actual. La següent comanda indica el proveïdor del servei GSM, en el nostre cas hauria de dir Movistar i la última, indica els bauds que pot treballar el mòdul.

El resultat de totes les operacions anteriors és aquest.



```

COM7 - Tera Term VT
File Edit Setup Control Window Help

IIII AT
OK
AT+CPIN=9812
OK
AT
OK
AT+CGMI
SIMCOM_Ltd
OK
AT+CPIN?
+CPIN: READY
OK
AT+CNUM
OK
AT+COPS?
+COPS: 0,0,"Movistar"
OK
AT+IPR=?
+IPR: (),(0,1200,2400,4800,9600,19200,38400,57600,115200)
OK

```

Figura 8.10. Resultat de diverses operacions amb el mòdul SIM900.

Cada cop que el mòdul encén la connexió a la xarxa, surten les quatre barres que es mostren a la primera fila, i a continuació respon correctament a la primera comanda AT. Se li afegeix el nombre PIN amb la comanda CPIN, a la CGMI respon SIMCOM_Ltd, a CPIN? Respon amb un OK i el que és curiós és que respongui també amb un OK quan se li demana pel nombre PIN. Teòricament hauria de donar-nos el nombre de telèfon i no pas un OK però això és degut a una falta de programació del mòdul. Si es vol el número, farà falta utilitzar altres comandes, però bé, tampoc és molt necessari fer-ho, al cap i a la fi això no és més que una prova per a veure si el mòdul funciona correctament.

Per últim, a la comanda de COPS? Ens dona "Movistar" com a companyia telefònica correctament i la última comanda ensenya tots els possibles baud rates als que pot treballar el mòdul.

Ja es veu que el mòdul funciona, la velocitat de resposta i comunicació sèrie és molt bona pel que ara ja es provarà a utilitzar amb un microprocessador i que sigui aquest el que demani les comandes AT.

8.2.2.2. Segona aplicació. Realització de trucades

Ara provarem a fer un programa per a poder trucar un telèfon mòbil amb el mòdul i Arduino per a veure que és capaç de connectar-se a la línia i realitzar la connexió. Per a fer aquest programa només seran necessàries tres comandes AT, les de trucar, penjar i introduir el PIN.

La part del loop quedaria per exemple d'aquesta manera:

```
void loop()
{
  SIM900.println("ATD617640982;");
  delay(20000);
  SIM900.println("ATH");
  delay(1000);
  while (1);
}
```

Amb aquest codi, es truca al mòbil després de la comanda ATD, s'esperen 20 segons a que s'efectui la trucada (de vegades tarda una estona a establir la connexió), pengem passat aquest interval i s'espera un segon per entrar en un bucle que no en sortirà per evitar que estigui constantment repetint el codi i per tant, trucant.

Ara bé, cal tenir en compte diferents factors de la placa:

Si s'utilitza els pins 7 i 8 per a la recepció i transmissió de dades, a part de posar els punts corresponents com a hardware, en el software caldrà també declarar-los com a pins de transmissió de dades amb la llibreria Software Serial.

Per altra banda, el mòdul no es connecta automàticament a la xarxa 2G, sinó que cal mantenir pitjat un botó per a fer-ho.

Sabent això, la funció setup tindrà un aspecte com aquest:

```
#include <SoftwareSerial.h>
SoftwareSerial SIM900(7, 8);

void setup()
{
  delay (5000);
  SIM900.begin(19200);
  Serial.begin(19200);
  delay (1000);
  SIM900.println("AT + CPIN = \"9812\"");
  delay(25000);
}
```

Es declara el port de comunicacions amb el mòdul amb el nom de SIM900 i es posa als pins 7 i 8. Quan s'inicialitza la placa, el primer que farà serà esperar un temps que se li dona per a encendre el mòdul. Després es configuren les velocitats del port del mòdul i del Arduino, i se li indica amb la comanda AT+CPIN el PIN de la targeta SIM de la placa. El retard final és perquè trobi una xarxa GSM.

Amb la unió d'aquestes dos parts de codi s'aconsegueix un programa capaç de trucar un cop al mòbil [74].

Cal explicar però que alimentant el mòdul amb el port USB del ordinador, la potència és insuficient per a fer que el mòdul es connecti a una xarxa i quan ho intenta els leds de la placa s'apaguen. El mateix passa amb una font de 5 V i 0.5 A. La manera que tenia més a mà per a alimentar amb un amperatge suficient el mòdul ha estat amb una pila de 9 V, i així el resultat és exitós com s'observa a la següent fotografia.



Figura 8.11. Fotografia captada en el moment de la trucada.

8.2.2.3. Tercera aplicació. Enviar SMS

Per a realitzar el programa d'enviament de SMS es necessita novament utilitzar la comanda per introduir el PIN de la targeta SIM i canviarem el de trucada per el d'enviar SMS.

Aquesta nova comanda però té algun pas més a seguir que l'anterior programa. Com que es tracta d'enviar un SMS predefinit, s'haurà d'escriure en algun moment després de la comanda AT+CMGF=1 al monitor sèrie del SIM900.

Així doncs, primer declararem el missatge en mode text amb la comanda AT+CMFG després s'escriurà la que permet enviar SMS, AT+CMGS. Aquesta, igual que la de fer una trucada, compta

amb la incorporació del número de telèfon al que se li enviarà el missatge. Quan se li ordena aquesta comanda, el mòdul posa un signe > al port sèrie que se li ha declarat per a que s'escrigui a continuació. Es deixarà un petit temps per a que el posi (sempre que hi hagi comunicació entre dos terminals s'haurà de deixar un petit marge de temps per a que realitzin la comunicació) i escriurem el missatge simplement al port, en aquest cas anomenat SIM900. Per a saber que ja s'ha acabat de escriure el missatge, el mòdul ho reconeix amb el caràcter control z o ctrl z pel que si anéssim al port i escrivíssim aquest caràcter, el missatge s'enviaria. Però això no ho podem fer amb software ja que no es tracta de cap nombre ni lletra, sinó de la combinació de dos tecles. Per a fer-ho per software, se sap que el caràcter que correspon al ctrl z és el número 26 pel que s'ha d'escriure (char) 26 per a simular prémer la combinació i que ho envii. Per acabar es deixarà un marge de temps de uns 4 segons per a que faci l'operació tot amb 2 s'ha comprovat que també funciona [74].

La funció quedaria doncs així:

```
void missatge_sms()
{
  SIM900.print("AT+CMGF=1\r");
  delay(100);
  SIM900.println("AT + CMGS = \"617640982\"");
  delay(100);
  SIM900.println("Arduino i modul SIM900 Vicenc Suriol 23/02/2107");
  delay(100);
  SIM900.println((char)26);
  delay(4000);
}
void loop()
{
  missatge_sms();
  while (1);
}
```

El resultat i comprovant de l'operació és aquest:

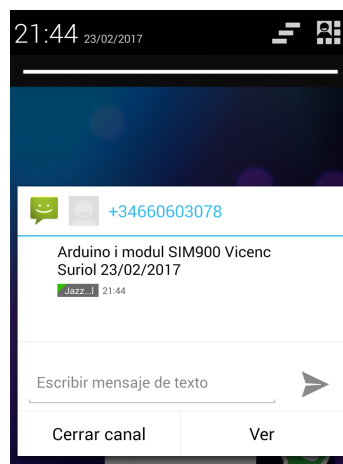


Figura 8.12. Instantània captada des del mòbil receptor del SMS enviat pel mòdul SIM900.

8.2.2.4. Quarta aplicació. Connectar-se a la xarxa.

Aquesta aplicació és ja més difícil de fer perquè inclou connectar-se a internet mitjançant la companyia telefònica i requereix entrar a la pàgina web de manera que el mòdul ho entengui.

Abans de començar cal detallar que fins ara totes les aplicacions dutes a terme s'han realitzat amb una targeta SIM de Movistar sense cap problema. Però a partir d'ara s'utilitzarà una SIM de Symio. Per a més informació sobre aquesta decisió, consultar l'apartat de selecció del material, al subapartat de transmissor de dades, allà s'explica el perquè la utilització d'ambdues targetes SIM.

Per a realitzar aquesta aplicació caldran diferents comandes AT que passem a esmentar a continuació:

Només començar, el mòdul haurà de connectar-se a la xarxa i definir el APN de la companyia telefònica igual que el seu usuari i contrasenya. Aquestes dades són fàcils de trobar per internet i en aquest cas (per a la companyia de Symio) el nom del punt d'accés és gprs-service.com i no té ni usuari ni contrasenya. Acte seguit, activarem el perfil de dades sense fils i obtindrem la nostra direcció IP.

AT+CGATT=1\r

AT+CSTT=\"gprs-service.com\",,\"

AT+CIICR

AT+CIFSR

CGATT s'utilitza per connectar-se o desconnectar-se del servei de domini de paquets o GPRS. CSTT es fa servir per a introduir el apn el nom d'usuari i la contrasenya. CIICR crea una connexió GPRS i CIFSR retorna la direcció IP local necessària per a connectar-se a la xarxa.

A continuació es fan servir ja les comandes per a rebre dades d'un servidor d'internet. Primer de tot cal començar una connexió TCP o HTTP, en aquest cas serà TCP amb AT+CIPSTART per a continuació utilitzat AT+CIPSEND per a enviar la direcció amb l'ordre de GET a la qual es vol connectar del servidor declarat a la comanda anterior per a realitzar la connexió i agafar tota la informació de la pàgina.

Abans de presentar l'esquema de programació donar crèdits del programa als programadors de Cooking-Hacks.com i de Prometec.net per a dissenyar una estructura robusta el qual m'ha servit per a facilitar-me l'estructura del codi. Per altra banda, el servidor al qual em connectaré és un creat per mi des de Heroku el qual parlarem més endavant a l'apartat del servidor.



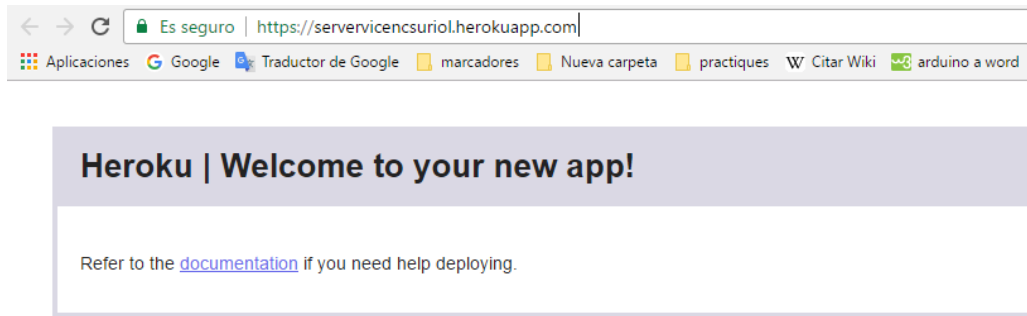


Figura 8.13. Servidor de prova creat per a realitzar el programa.

```
#include <SoftwareSerial.h>
SoftwareSerial SIM900(7, 8);

char aux_str[50];

char direccio[] = "GET // HTTP/1.1\r\nHost:
servivicencsuriol.herokuapp.com\r\nConnection:close\r\n\r\n";
```

Aquesta és la direcció a la qual es connectarà el mòdul. Cal dir que Arduino no suporta pàgines web amb capa de seguretat SSL el que vol dir que no es poden posar pàgines HTTPS. Per sort Heroku admet ambdós protocols, HTTP i HTTPS. L'estructura de la direcció serà sempre similar amb la necessitat de afegir elements com `\r\n` que amb C signifiquen un enter.

```
void setup()
{
  SIM900.begin(19200);
  Serial.begin(19200);
  delay(1000);
  iniciar();
}

void loop()
{
  if (Serial.available())
  {
    switch (Serial.read())
    {
      case 'h':
        PeticioHttp();
        break;
    }
  }
  if (SIM900.available()) {
    Serial.write(SIM900.read());
  }
}
```

Aquesta és l'estructura del programa, al Setup farem les instruccions necessàries per a connectar-se a la xarxa GPRS i al Loop farem les que es connecten al servidor i agafen les dades. Totes ja vistes abans.

Quan ja s'hagi llegit les dades del servidor, si el port es troba disponible, escriurà tot el que llegeixi del servidor el mòdul per a que ho puguem veure pel monitor sèrie.

Per a que el programa sigui robust, s'afegeix una funció que es seguirà cada vegada que s'introdueixi una comanda AT anomenada enviarAT. Es declararà quina d'elles, la resposta que s'espera i el temps que ha d'esperar en rebre la resposta el programa. Veiem línies de codi que netegen el buffer d'entrada, l'enviament de la comanda AT pel port sèrie, una que llegeix la resposta mentre el mòdul envia dades pel port sèrie, comprova que la resposta rebuda sigui igual a la esperada i mentre no ho sigui que vagi passant el temps fins a rebre la resposta correcta [75].

```
void iniciar()
{
    enviarAT("AT+CPIN=\"9103\"", "OK", 1000);
    Serial.println("");
    delay (5000);

    while ( enviarAT("AT+CREG?", "+CREG: 0,1", 1000) == 0 )
    {
    }
    enviarAT("AT+CGATT=1\r", "OK", 1000);
    enviarAT("AT+CSTT=\"gprs-service.com\",,,\", \"OK\", 3000);
    enviarAT("AT+CIICR", "OK", 3000);
    enviarAT("AT+CIFSR", "", 3000);
}
```

Aquesta funció es fa un cop cada vegada que s'encén el mòdul, i posa el PIN, i fa les primeres comandes ja explicades. AT+CREG? Pregunta per el desbloqueig de la SIM i que el mòdul estigui ja preparat per ser operatiu.

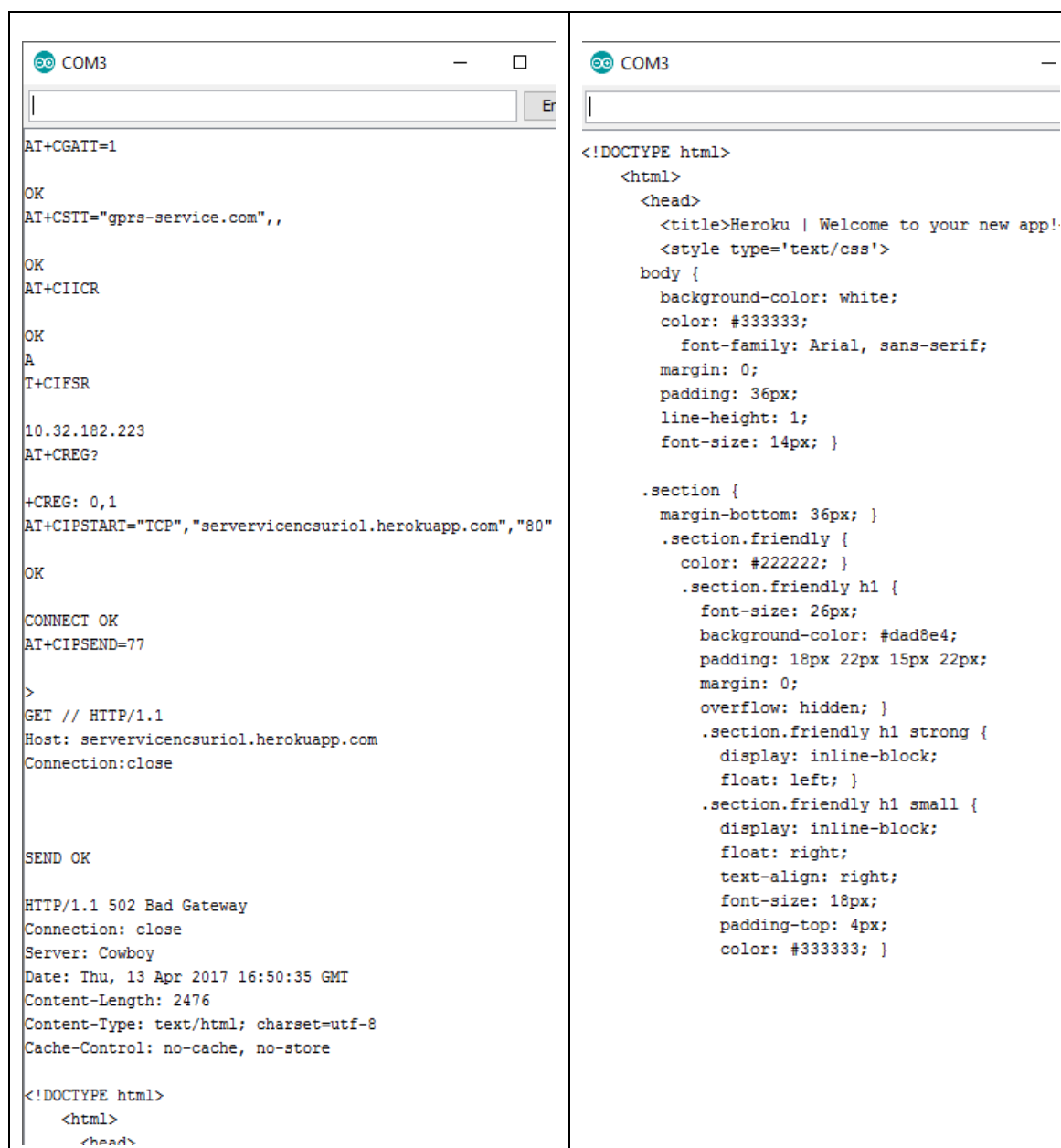
```
void PeticioHttp()
{
    if (enviarAT("AT+CREG?", "+CREG: 0,1", 1000) == 1) {
        enviarAT("AT+CIPSTART=\"TCP\", \"servicencsuriol.herokuapp.com\", \"80\", \"CONNECT OK\", 5000);
        sprintf(aux_str, "AT+CIPSEND=%d", strlen(direccio));
        if (enviarAT(aux_str, ">", 10000) == 1)
        {
            enviarAT(direccio, "OK", 10000);
        }
    }
}
```

La funció `PeticioHttp` comprova la connexió a la xarxa de nou i envia el CIPSTART amb TCP com a connexió i el servidor al que es vol connectar. Quan s'hagi connectat, enviarà dades a través de TCP amb la direcció esmentada al principi.

Quan s'envia una direcció URL amb el mòdul, és essencial definir la longitud de caràcters de la pròpia URL. Amb la línia de `sprintf` s'ajunta a la variable `aux_str` la comanda `AT+CIPSEND` que es mostra i que serveix per a enviar una operació, amb la longitud de caràcters de la direcció URL i que això es fa amb la comanda `strlen` (direcció) que donarà un decimal i per això el `=%d`. Ara doncs no enviarem una comanda AT pel port, sinó que s'enviarà la variable `char aux-str` que en sí ja conté una comanda AT, la que farà enviar amb justament la longitud de caràcters com ho vol el mòdul.

Sabem que acte seguit d'enviar aquesta comanda, el mòdul retornarà un signe `>` i si és així enviarem la direcció a la que volem accedir d'una vegada.

El programa no és fàcil, i de fet hi ha hagut molts problemes per a que funcionés bé. Però al final el resultat és el que es mostra a continuació.



```

COM3
AT+CGATT=1
OK
AT+CSTT="gprs-service.com",,
OK
AT+CIICR
OK
A
T+CIFSR
10.32.182.223
AT+CREG?
+CREG: 0,1
AT+CIPSTART="TCP","servivicencsuriol.herokuapp.com","80"
OK
CONNECT OK
AT+CIPSEND=77
>
GET // HTTP/1.1
Host: servivicencsuriol.herokuapp.com
Connection:close

SEND OK

HTTP/1.1 502 Bad Gateway
Connection: close
Server: Cowboy
Date: Thu, 13 Apr 2017 16:50:35 GMT
Content-Length: 2476
Content-Type: text/html; charset=utf-8
Cache-Control: no-cache, no-store

<!DOCTYPE html>
<html>
<head>

```

```

COM3
<!DOCTYPE html>
<html>
<head>
<title>Heroku | Welcome to your new app!
<style type='text/css'>
body {
background-color: white;
color: #333333;
font-family: Arial, sans-serif;
margin: 0;
padding: 36px;
line-height: 1;
font-size: 14px; }

.section {
margin-bottom: 36px; }
.section.friendly {
color: #222222; }
.section.friendly h1 {
font-size: 26px;
background-color: #dad8e4;
padding: 18px 22px 15px 22px;
margin: 0;
overflow: hidden; }
.section.friendly h1 strong {
display: inline-block;
float: left; }
.section.friendly h1 small {
display: inline-block;
float: right;
text-align: right;
font-size: 18px;
padding-top: 4px;
color: #333333; }

```

Figura 8.14. Monitor Sèrie amb les dades rebudes.

La primera imatge mostra tot el procés de connexió el qual passa el mòdul fins a arribar a la petició del servidor ja explicat fins ara. Podem veure un SEND OK que indica que la petició s'ha fet amb èxit i a continuació el servidor envia tota la seva informació que és llegida a continuació pel mòdul i Arduino. Tot i que visualment només es veuen unes quantes frases al entrar al servidor, realment a informació que comprèn és molt més àmplia, i això ho podem veure a la segona imatge. Una de les primeres línies posa Heroku Welcome to your new app com a Title que és justament el que té el servidor en aquest moment i per tant es pot assegurar que el mòdul s'ha connectat a ell perfectament.

8.3. Circuit d'alimentació i càrrega

El mòdul GPRS que s'ha estat provant presenta pics de fins a 2A que no ha estat capaç de subministrar una sola bateria per que s'ha adquirit una altra del mateix model. Amb dues bateries en paral·lel, l'amperatge que és capaç de subministrar el model es doble deixant el voltatge a 4.8 V.

Amb dues el procediment s'ha pogut dur a terme. Per altra banda, així es guanya amb vida útil sense recàrrega.

El dispositiu però, no hauria d'anar directament connectat a la bateria, a més que ara mateix no hi ha manera d'alimentar el circuit. Aquest és l'objectiu de l'apartat, el de crear un circuit d'alimentació pel dispositiu que es pugui carregar amb el mòdul de càrrega adquirit.

L'esquema que s'ha pensat ha estat el següent:

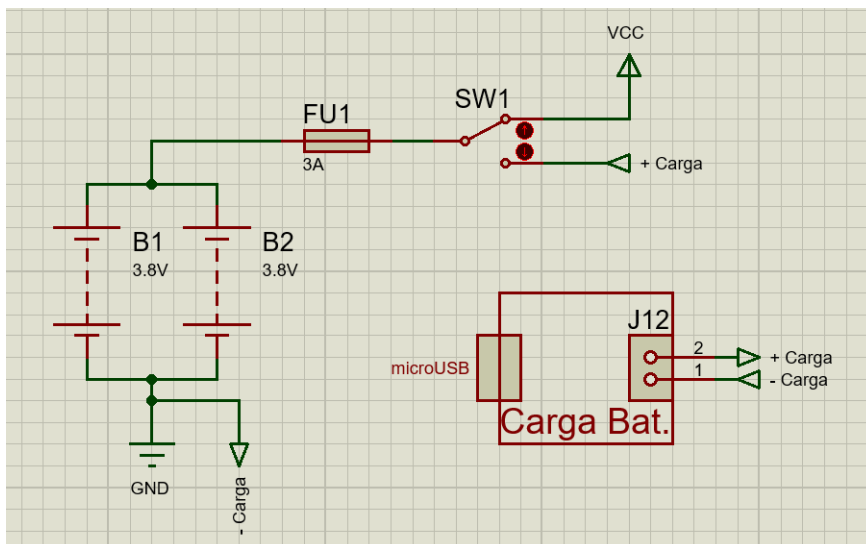


Figura 8.15. Esquema electrònic del circuit d'alimentació.

En ell tenim les dues bateries en paral·lel, amb el positiu connectat directament a un fusible de 3 A (el mòdul té puntes de fins a 2 A i el microprocessador consumeix uns 400 mA) que està una mica sobredimensionat per a evitar sobrecorrents i cremar el circuit de qualsevol possible averia. Després del fusible hi ha un botó d'ancoratge que permet alternar entre alimentar tot el circuit, o donar càrrega a les bateries. És així per evitar simplement que s'amorri el carregador en el cas que es tanqués el circuit alimentant per aquest. O la bateria dona subministrament al circuit, o es carrega la bateria.

El circuit s'ha dut a terme a una placa i aquest n'ha estat el resultat:

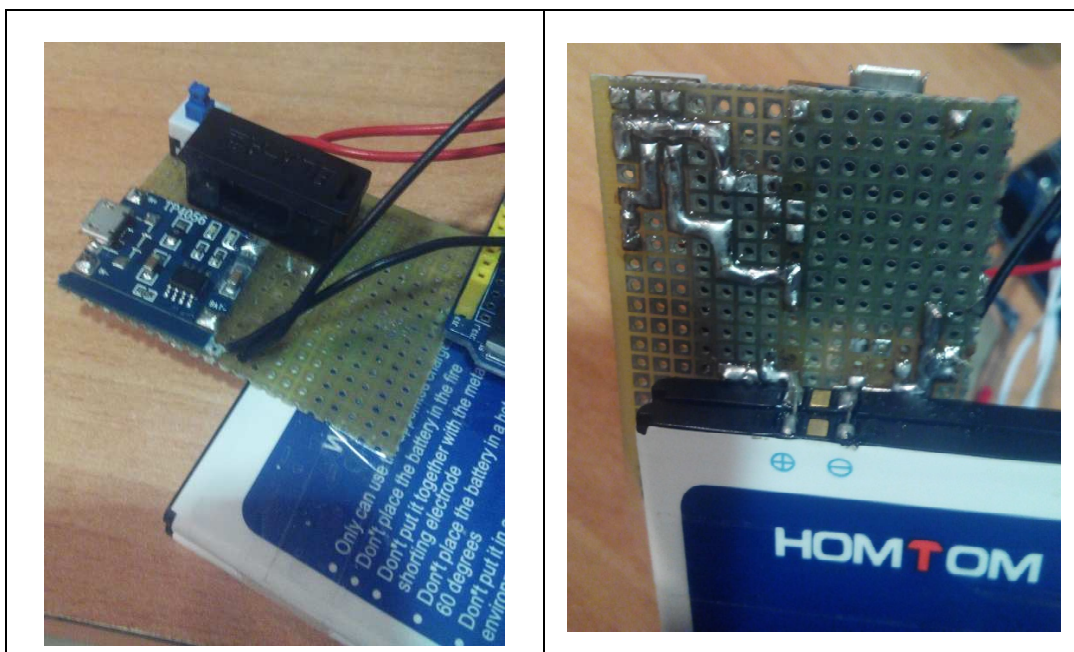


Figura 8.16. Placa amb els components soldats.

El mòdul carrega la bateria correctament i quan es canvia de posició el botó hi ha tensió entre els extrems del punt VCC i GND.

Hi ha dos parells de cables d'alimentació, un per Arduino i el receptor i l'altre per el mòdul SIM900.

9. Creació del Servidor

Per a poder fer funcionar tot el projecte, un servidor és un element indispensable que cal crear abans de programar qualsevol aplicació amb Arduino ja que aquest estarà inclòs en totes les programacions següents. La funció doncs del servidor és, per una banda, comunicar Arduino amb el núvol per a penjar-hi les dades, i per altra, que l'aplicació mòbil s'hi connecti per a agafar les dades, mostrar-les i interpretar-les correctament. La tasca tot i que fàcilment explicada, dur-la a terme no és gens fàcil, en aquest apartat es veurà tot el procés.

El servidor es fa amb Heroku com ja s'ha explicat en apartats anteriors, però tot servidor s'ha de programar. La programació d'aquesta interfície no es fa des d'internet, sinó que s'ha de pujar des d'un ordinador amb comandes del símbol del sistema de Windows, més conegut com a consola de comandes o CMD. A més el servidor es pot programar amb diferents llenguatges, php i python essent els més utilitzats. Per a tot aquest procés, Heroku posa al abast dels seus usuaris unes eines per a poder pujar els seus projectes locals al servidor anomenat Heroku Toolbelt.

9.1. CLI Toolbelt i fitxers pel servidor de Heroku

Per a programar el servidor cal baixar-se el Command line interface o Heroku Toolbelt. Un conjunt d'eines de CMD de Windows per a pujar programes al servidor per a què funcioni com es vulgui.

Cal un cop instal·lat CLI Toolbelt, elegir una carpeta local on començar a posar els fitxers que serviran per a que el servidor entengui la programació i executi les comandes ordenades. S'ha escollit Python com a llenguatge de programació del servidor i per a començar, cal afegir a la nova carpeta que inclourà els arxius alguns estàndard per a Heroku. Aquests són els següents:

- Carpeta .Git.
Git és realment un software aliè a Heroku que s'utilitza per a pujar els projectes al núvol i controlar les versions del software quan aquest té un cert nombre d'arxius com, per exemple aquest servidor. Per tant doncs no té res a veure amb el servidor en sí, però és l'eina utilitzada per a poder enviar software desenvolupat en local a Heroku.
- Fitxer .gitignore.
Utilitzat per Git també, es tracta d'un fitxer de .txt que com el nom diu s'ha d'ignorar. Si s'obra es pot observar una mica de codi poc clar.
- Fitxer procfile.

Aquest fitxer ja és més important i conté la informació que li indica a Heroku què ha d'executar per a iniciar l'aplicació. Si s'obre es veurà una sola línia de codi on hi destaca la paraula `app`: `app` referent a que executi aquest fitxer que després passarem a veure.

- Fitxer `procfile.windows`.

El nom ja indica en sí que es similar al anterior. La seva tasca és la mateixa però s'utilitza quan s'executa el servidor en local. Per això hi ha el `.windows` a més. Cal explicar que si l'obrim veurem un path que direcciona a la `C:/` del nostre ordinador, a `python33` i `python.exe`. Si el nostre programa de python no es troba en aquesta direcció, s'ha de modificar per la corresponent. Més endavant ja es parlarà de com es fa per a fer anar el servidor.

- Document de text `requirements`.

En aquesta llista en format `.txt` s'inclouen totes les llibreries i paquets de python que Heroku ha d'utilitzar i instal·lar al servidor per a que aquest funcioni correctament. Entre ells destaquen `SQLAlchemy`, `Flask` o `HTTPauth` per es volgués demanar un registre per a entrar en alguna pàgina del servidor.

- Document de text `runtime`.

`Runtime` indica a Heroku quin llenguatge de programació i quina versió ha de compilar per a poder pujar el servidor bé. En aquest cas, `Python 3.5.1`.

Tots aquest fitxers han estat distribuïts per a un professor de la universitat que també m'ha ajudat a entendre com programar el servidor des de un ordinador personal.

9.2. La Base de dades

El primer que es programarà serà la base de dades. El més usual en la creació d'una base de dades és fer-la local amb programes com `PostgreSQL` però realment aprendre a utilitzar un nou software per a després refer la tasca per a pujar-la a un servidor no em val massa la pena al ser una BD força simple.

Una base de dades està composta per tres elements principals: Entitats, objectes diferenciadors entre ells, per exemple usuaris o missatges. Atributs, característiques de les entitats, és el contingut d'elles. Per exemple la hora del missatge, o el nom del usuari. Per últim la relació entre els atributs. El vincle entre les entitats per exemple cada usuari té missatges. Aquesta relació seria del tipus 1:n on 1 representa 1 usuari i n representa nombre indefinit de missatges. També hi ha relacions 1:1 o n:m depenent de la BD i com es relacionen les seves entitats.

Ara mateix, la nostra BD necessita emmagatzemar Coordenades i ja està. Podríem pensar en afegir usuaris, i que cada usuari tingui les seves coordenades, però realment només existeix o existirà un dispositiu que enviarà dades al servidor, pel que no té sentit afegir una entitat d'usuaris. Per tant

doncs, necessitem d'una entitat que s'anomeni Coordenades amb tres atributs, Latitud i Longitud com a principals, i la data i hora faran el tercer i últim d'ells. Si hi haguessin usuaris, cada coordenada tindria els seus atributs i estaria dins d'un usuari, on cada un d'ells tindria moltes coordenades fent una relació de 1:n.

El diagrama que pot representar la BD pot ser un com aquest:

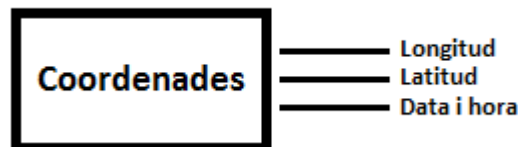


Figura 9.1. Model entitat-relació de la BD .

L'única entitat que tindrà la BD presenta tres atributs i, clarament, cap relació al no haver-hi més entitats.

9.2.1. Programació de la Base de Dades

Es necessita crear un arxiu.py que crearà el funcionament de la base de dades que tenim pensat.

Les entitats es defineixen com a Class (db.Model) i se'ls hi indica els seus atributs amb columnes i declaren de quin tipus seran els valors que s'hi introdueixin. Amb aquest codi com a model:

```
class Mensajes(db.Model):
    __tablename__ = 'mensajes'

    id = db.Column(db.Integer, primary_key=True)
```

Es pot pensar que l'estructura de la classe seria així:

```
import datetime
class Coordenades(db.Model):
    __tablename__ = 'coordenades'

    id = db.Column(db.Integer, primary_key=True)
    latitud = db.Column(db.String(200))
    longitud = db.Column(db.String(200))
    temps = db.Column(db.DateTime)
```

S'ha afegit un import per a que el servidor pugui agafar la seva data i hora directament. Tot i que l'atribut id no fa falta, tampoc fa nosa i potser en un futur alhora de agafar les dades per l'aplicació aquest id facilitarà la feina de quines agafar.

Ara ja s'ha declarat com és la BD però no d'on recaptarà les dades. Això s'ha de declarar a continuació i s'ha fet d'aquesta manera:

```
def __init__(self, latitud, longitud):
    self.latitud = latitud
    self.longitud = longitud
    self.tiempo = datetime.datetime.now()

def __repr__(self):
    return % (self.latitud, self.longitud)
```

`def __init__` és com amb Python es construeix una classe, i el `self` representa `Coordenades`. Python demana que aquesta sigui l'estructura. A continuació es declara d'on es treurà la informació, en aquest cas, `latitud` de la BD s'extrau de la variable `latitud` que ja veurem on es troba més endavant, i igual amb la resta.

`Def __repr__` retorna una representació impresa del objecte, en aquest cas `Coordenades`. En aquest cas retornarà `latitud` i `longitud` de la BD.

Ja estaria la BD creada. Però farà falta un altre programa per a crear les taules definides en el codi anterior. Només faria falta executar-lo una vegada i ha d'importar l'arxiu creat ara, i la instrucció:

```
db.create_all()
```

Es deix el codi sencer de la BD i del servidor a continuació, al Annex de la memòria per a evitar afegir molt de codi en aquests apartats tenint en compte que ja s'ha comentat tot el fonamental del codi.

9.3. Programació del Servidor

Si bé s'han adjuntat tots els fitxers necessaris per a pujar el servidor a Heroku i que aquest l'interpreti correctament, ara falta programar-lo en funció de com es vulgui que funcioni. Per a fer-ho necessitem de més d'un codi.

Primer de tot fa falta crear l'aplicació Flask. Flask és un marc de referència per a servidors web utilitzat per a principiants perquè és més aviat fàcil d'utilitzar. El nostre servidor ha de tenir una base de dades també, pel que s'ha d'obrir una connexió amb ella. Aquest codi que s'ha anomenat `inicia.py`, detecta si l'app es troba en local o en Heroku i ajusta diferents paràmetres per a què funcioni igualment. Al Annex es pot trobar el codi.

Passem doncs al codi important del servidor. Ara es veurà què ha de fer el servidor cada cop que s'entra en ell d'alguna manera.

Com que Arduino entra al servidor mitjançant un GET, s'ha de declarar que quan ho faci, el servidor guardi les dades a la BD. El primer que s'ha de fer doncs és guardar a unes variables el que arriba després de la URL del servidor, que tindrà un aspecte d'aquest estil:

?lat=xx.xxx&lon=yy.yyy

Un cop enregistrades la lat i la lon, les guardarem en una variable que anomenem msj i ho pujarem a la BD d'aquesta manera per a què ho pugui entendre el programa creat abans.

En definitiva el codi queda així:

```
class ArduinoAPI(Resource):  
  
    def get(self):  
  
        latitud = request.args.get('lat')  
        longitud = request.args.get('lon')  
        msj = Mensajes(latitud, longitud)  
        db.session.add(msj)  
        try:  
            db.session.commit()  
            return {'result': True}  
        except:  
            db.session.rollback()  
            return {"error": "Error"}, 404
```

El try except, és un codi per a que si la pujada ha tingut èxit, retorni per pantalla result:True, i si no ha estat el cas, que retorni error i reverteixi els canvis.

Si tot va bé el programa ja hauria de guardar les dades enviades per Arduino a un espai al núvol declarat també en aquest arxiu, que per cert, s'anomena app.py disponible per complert al Annex. Per a declarar aquest espai s'ha d'afegir al codi aquesta línia.

```
api.add_resource(ArduinoAPI, '/msgs/api/v1.0/arduino', endpoint='arduino')
```

Ja s'ha completat l'enllaç del hardware al núvol, però el servidor també necessitarà mostrar les dades quan se li faci un GET des de l'aplicació.

Per a fer aquesta segona part, s'ha de crear un altre punt d'accés que anomenarem AppAPI ja que no es permet fer un get diferent en un mateix punt d'accés com és evident. Tot i així, es podria programar de tal manera que si rep coordenades les guardi, i si en canvi no rep res o rep alguna cosa diferent, mostri la informació, però per evitar ser poc ordenats i estalviar-nos possibles errors entre les dues interfícies, Arduino i l'aplicació tindran un espai del servidor reservat per ambdues.

Volem doncs que quan l'aplicació faci un GET a AppAPI, mostri les últimes coordenades rebudes. Quantes exactament? Bé ho farem amb 20 però seria només canviar un nombre.

El problema d'això, és que les dades que interessin són les últimes enregistrades, i no pas les primeres pel que primer s'haurà d'ordenar-les per orde descendent en funció del temps, i s'han de limitar a 20. Si això es guarda a una variable msjs i es fa un Return de la latitud, de la longitud i del temps d'aquests msjs s'obtindrà la taula desitjada. Per a realitzar aquest codi m'he basat amb aquesta línia dissenyada pel meu professor d'informàtica:

```
msjs = Mensajes.query.all()
return [{'texto':m.texto, 'tiempo':str(m.tiempo),
        'autor':m.autor.nombre} for m in msjs]
```

El qual entrava a la base de dades de Mensajes i els agafava tots en una llista, i d'aquests retornava el text, el temps, l'autor.

El resultat final després de forces intents i cerca per internet ha estat el següent:

```
# GESTIÓ app
class AppAPI(Resource):

    def get(self):
        msjs =
        Mensajes.query.order_by(desc(Mensajes.tiempo)).limit(20).all()
        return [{'id':m.id, 'latitud':m.latitud,
                'longitud':m.longitud, 'tiempo':str(m.tiempo)} for m in msjs]

    def put(self):
        pass

    def post(self):
        pass
```

El que més interessa és com s'ha guardat la llista de coordenades a la variable msjs, amb ordre descendent en funció de Mensajes.tiempo i amb un límit de 20 i no amb el *all* que hi havia abans.

El return és similar variant el que retorna. S'ha afegit la id però realment no seria necessari. El put i el post no es vol que faci res el servidor, pel que si arriba algun cop un put o un post, el servidor no farà res enlloc de donar error. Però no crec que algú li demani mai alguna d'aquestes funcions.

Un cop dissenyat el codi, fa falta pujar-lo a Heroku amb les eines que s'han baixat abans, però cal detallar-ne el procediment perquè la tasca no és senzilla.

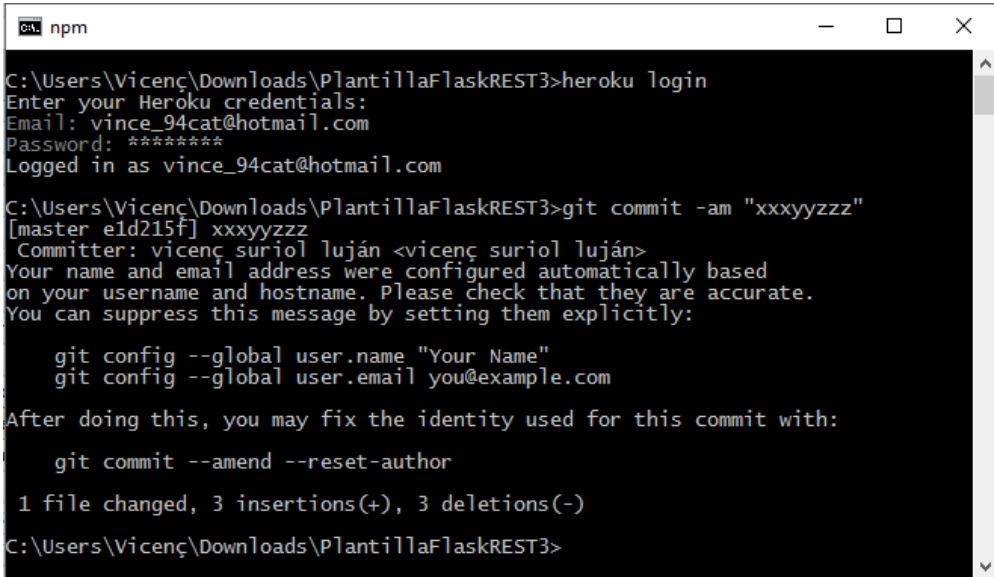
Caldrà obrir una finestra de comandes a la carpeta que contingui tots els arxius i escriure-hi Heroku login. Si Heroku toolbelt s'ha instal·lat correctament, veurem com podem introduir el nostre usuari i contrasenya d'Heroku.

Registrats, caldrà especificar a quin servidor es vol pujar l'aplicació, sinó ho farà al servidor utilitzat anteriorment si realment s'ha creat un, sinó donarà error. Ara s'ha de posar la comanda:

```
git commit -am "xxxxyzzz"
```

La qual obliga a canviar els arxius i a modificar els que hi haguessin al servidor d'Heroku.

El resultat hauria de ser com es mostra a la següent imatge:



```
npm
C:\Users\Vicenç\Downloads\PlantillaFlaskREST3>heroku login
Enter your Heroku credentials:
Email: vince_94cat@hotmail.com
Password: *****
Logged in as vince_94cat@hotmail.com

C:\Users\Vicenç\Downloads\PlantillaFlaskREST3>git commit -am "xxxxyzzz"
[master e1d215f] xxxxyzzz
Committer: vicenç suriol luján <vicenç suriol luján>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 3 insertions(+), 3 deletions(-)

C:\Users\Vicenç\Downloads\PlantillaFlaskREST3>
```

Figura 9.2. Finestra de comandes amb les instruccions descrites .

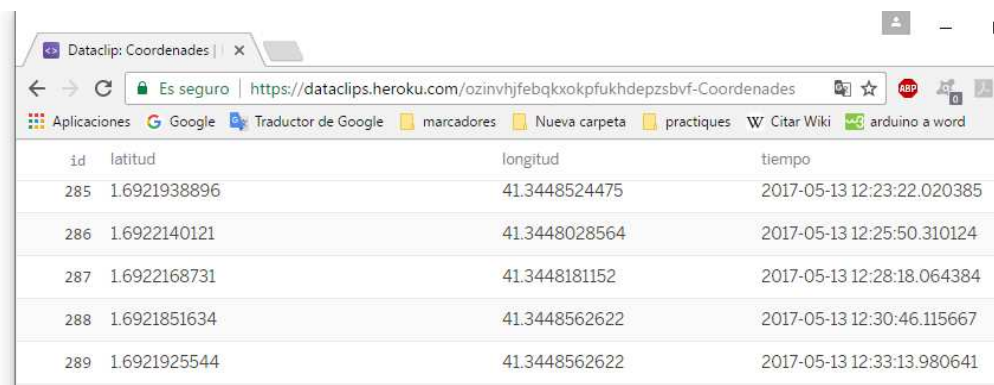
Les modificacions que s'ha fet en aquest cas han estat tres i han involucrat un arxiu, exactament el app.py.

Ara ja només s'ha de penjar-ho a Heroku amb la instrucció:

```
GIT push Heroku master
```

La qual tardarà uns dos minuts en donar efecte i ja es podran provar els nous canvis en el servidor.

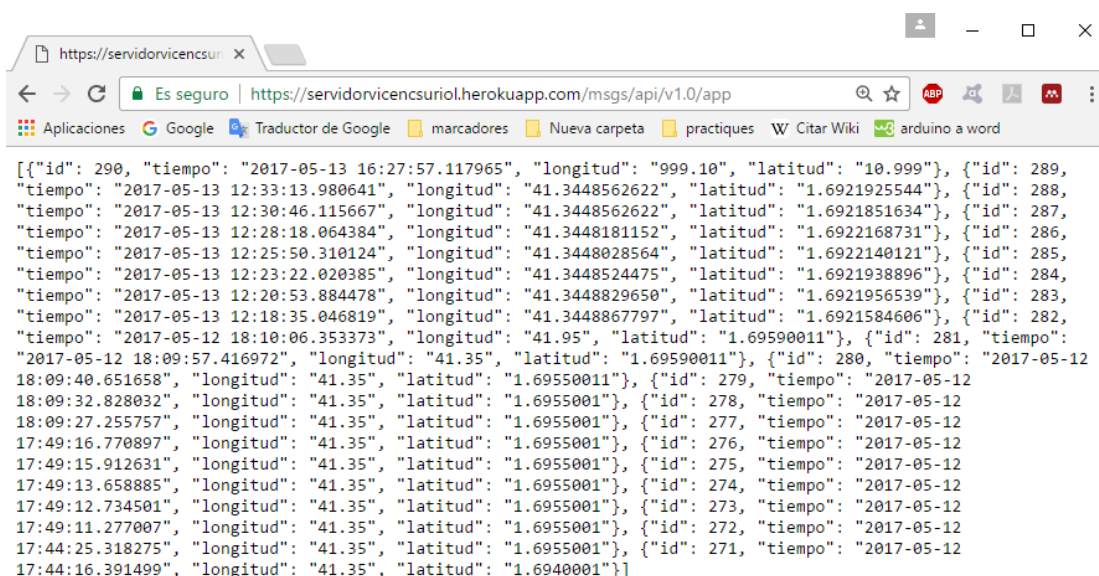
Ara es comprovarà que realment tot funciona correctament. Cal dir però que realment fer funcionar el servidor en el seu funcionament òptim ha estat una feina que ha durat molt de temps, més del previst al diagrama de Gaant. Tot i així es mostrarà el resultat final a continuació.



id	latitud	longitud	tiempo
285	1.6921938896	41.3448524475	2017-05-13 12:23:22.020385
286	1.6922140121	41.3448028564	2017-05-13 12:25:50.310124
287	1.6922168731	41.3448181152	2017-05-13 12:28:18.064384
288	1.6921851634	41.3448562622	2017-05-13 12:30:46.115667
289	1.6921925544	41.3448562622	2017-05-13 12:33:13.980641

Figura 9.3. Finestra d'un navegador amb la BBDD.

Es pot observar la Base de Dades en la primera imatge, un seguit de coordenades que en el següent apartat es veurà com les ha enviat Arduino. A la següent captura veurem el que veurà l'aplicació, totes les coordenades en una llista que podrà ser llegida fàcilment per a poder posar les coordenades en un mapa.



```
[{"id": 290, "tiempo": "2017-05-13 16:27:57.117965", "longitud": "999.10", "latitud": "10.999"}, {"id": 289, "tiempo": "2017-05-13 12:33:13.980641", "longitud": "41.3448562622", "latitud": "1.6921925544"}, {"id": 288, "tiempo": "2017-05-13 12:30:46.115667", "longitud": "41.3448562622", "latitud": "1.6921851634"}, {"id": 287, "tiempo": "2017-05-13 12:28:18.064384", "longitud": "41.3448181152", "latitud": "1.6922168731"}, {"id": 286, "tiempo": "2017-05-13 12:25:50.310124", "longitud": "41.3448028564", "latitud": "1.6922140121"}, {"id": 285, "tiempo": "2017-05-13 12:23:22.020385", "longitud": "41.3448524475", "latitud": "1.6921938896"}, {"id": 284, "tiempo": "2017-05-13 12:20:53.884478", "longitud": "41.3448829650", "latitud": "1.6921956539"}, {"id": 283, "tiempo": "2017-05-13 12:18:35.046819", "longitud": "41.3448867797", "latitud": "1.6921584606"}, {"id": 282, "tiempo": "2017-05-12 18:10:06.353373", "longitud": "41.95", "latitud": "1.69590011"}, {"id": 281, "tiempo": "2017-05-12 18:09:57.416972", "longitud": "41.35", "latitud": "1.69590011"}, {"id": 280, "tiempo": "2017-05-12 18:09:40.651658", "longitud": "41.35", "latitud": "1.69550011"}, {"id": 279, "tiempo": "2017-05-12 18:09:32.828032", "longitud": "41.35", "latitud": "1.69550011"}, {"id": 278, "tiempo": "2017-05-12 18:09:27.255757", "longitud": "41.35", "latitud": "1.69550011"}, {"id": 277, "tiempo": "2017-05-12 17:49:16.770897", "longitud": "41.35", "latitud": "1.69550011"}, {"id": 276, "tiempo": "2017-05-12 17:49:15.912631", "longitud": "41.35", "latitud": "1.69550011"}, {"id": 275, "tiempo": "2017-05-12 17:49:13.658885", "longitud": "41.35", "latitud": "1.69550011"}, {"id": 274, "tiempo": "2017-05-12 17:49:12.734501", "longitud": "41.35", "latitud": "1.69550011"}, {"id": 273, "tiempo": "2017-05-12 17:49:11.277007", "longitud": "41.35", "latitud": "1.69550011"}, {"id": 272, "tiempo": "2017-05-12 17:44:25.318275", "longitud": "41.35", "latitud": "1.69550011"}, {"id": 271, "tiempo": "2017-05-12 17:44:16.391499", "longitud": "41.35", "latitud": "1.69400011"}]
```

Figura 9.4. Finestra d'un navegador amb la URL que fa servir l'aplicació.

10. Disseny del codi d'Arduino

Si ve s'ha comprovat que tot anés correctament i provat el funcionament dels mòduls, en aquest apartat, a part de millorar les parts del codi dissenyades a l'apartat de posada en marxa i que han fet de base fins ara per el projecte, s'acoblaran totes elles i es faran funcionar conjuntament per a poder donar per enllestida la part de programació del dispositiu. S'ha creat un diagrama de flux per a entendre el funcionament del codi en la seva totalitat abans d'explicar punt per punt com s'ha fet cada cosa.

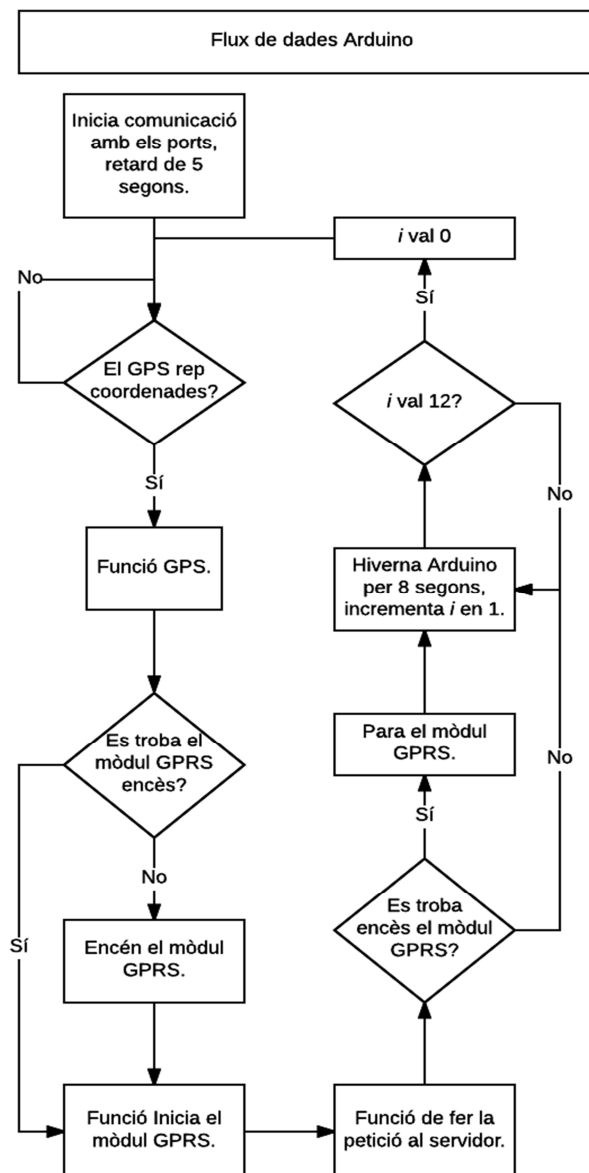


Figura 10.1. Diagrama de flux del codi d'Arduino.

10.1. Software del GPS

Vàrem deixar el codi un cop aconseguit mostrar per pantalla tres de les variables més importants per al projecte: El nombre de satèl·lits fixes, la latitud del receptor i la seva longitud.

Però cal fer-se la pregunta de si aquest codi és prou robust com per després poder ser acoblat i engrandit i seguir funcionant perfectament o almenys, bé. Una bona manera de saber si un programa podrà ser ampliat i seguir presentant els mateixos resultats és afegint-hi un temps d'espera que simularà altres operacions i línies de codi que estarà fent el microcontrolador quan no es trobi en la funció de extracció de dades del NMEA que li envia el receptor. Això es fa fàcilment amb la instrucció següent:

```
delay ( Temps ) ;
```

On Temps correspon al valor en milisegons que valdrà el retard o temps d'espera.

La pregunta és doncs: Segueix el codi mostrant pel monitor sèrie les dades desitjades si es posa per exemple un petit temps d'espera de mig segon?

La resposta (havent fet la prova per suposat) és que no.

A mesura que es va augmentant el temps de 1 milisegon a 1 segon, el refresc de dades pel monitor es va fent més lenta fins arribar als 15 segons de mitjana per a un temps d'espera de 500 ms. Si es va augmentant fins a 1 segon, ja no es refresca cap dada.

Per a poder solucionar-ho primer de tot, s'ha de saber perquè passa això i quina part del codi és la que ha deixat de funcionar com abans al afegir-hi un temps d'espera a la funció del loop, just després de la crida de la funció GPS i abans de tancar-la.

El codi té dues parts ben separades les quals val la pena focalitzar. La primera, que consta de només dues línies de codi que són les següents:

```
while(portGPS.available() )
{
  gps.encode(portGPS.read() );
}
```

Entra al bucle si el port on s'ha connectat el receptor està enviant dades. Si és així, les llegeix i guarda en una part de la memòria.

Un dels inconvenients d'utilitzar llibreries és que no pots saber en detall quina funció realitza cada instrucció específica d'aquesta tot i que pots investigar-ho mirant com es comporta imprimint-la pel monitor.

La segona part només imprimeix les dades si les dades han estat actualitzades en comparació a les anteriors que tenia guardades. Pel que pensant-ho bé, és molt probable que el problema es trobi a la primera part.

Si treus la instrucció `if(gps.location.isUpdated())`, resulta que el codi ja es torna a actualitzar amb el temps d'espera que se li posi independentment d'aquest, però ho fa molt més ràpid que de costum i tots els valors són iguals, i si el temps és superior al segon, tots els valors són zeros. Amb aquesta informació podem veure que el codi es topa amb la instrucció del `if()` quan li pregunta si està actualitzada, com sempre és que no, el codi no es mostra pel monitor.

Sabent que no s'enregistren noves dades, es molt probable que el microcontrolador no entri a la funció `while()` de la primera part perquè sinó aniria guardant noves dades. Això pot ser degut a que el moment en què es troba `available()` el portGPS sigui un petit interval de temps en comparació amb el temps total. D'aquesta manera, amb un retard de temps, si la funció ha de coincidir que s'estigui rebent dades pel port quan faci la instrucció i no ho aconsegueix, haurà de passar una altra vegada per el retard de temps abans de tornar-ho a intentar.

Saber si aquest és el problema és fàcil si imprimim aquesta instrucció pel monitor sèrie. Exactament amb aquesta instrucció:

```
Serial.print(portGPS.available());
```

Posada a la funció `loop()`, podem veure quin valor té quan rep senyal i quan no tant amb un retard de temps com en el comportament normal.

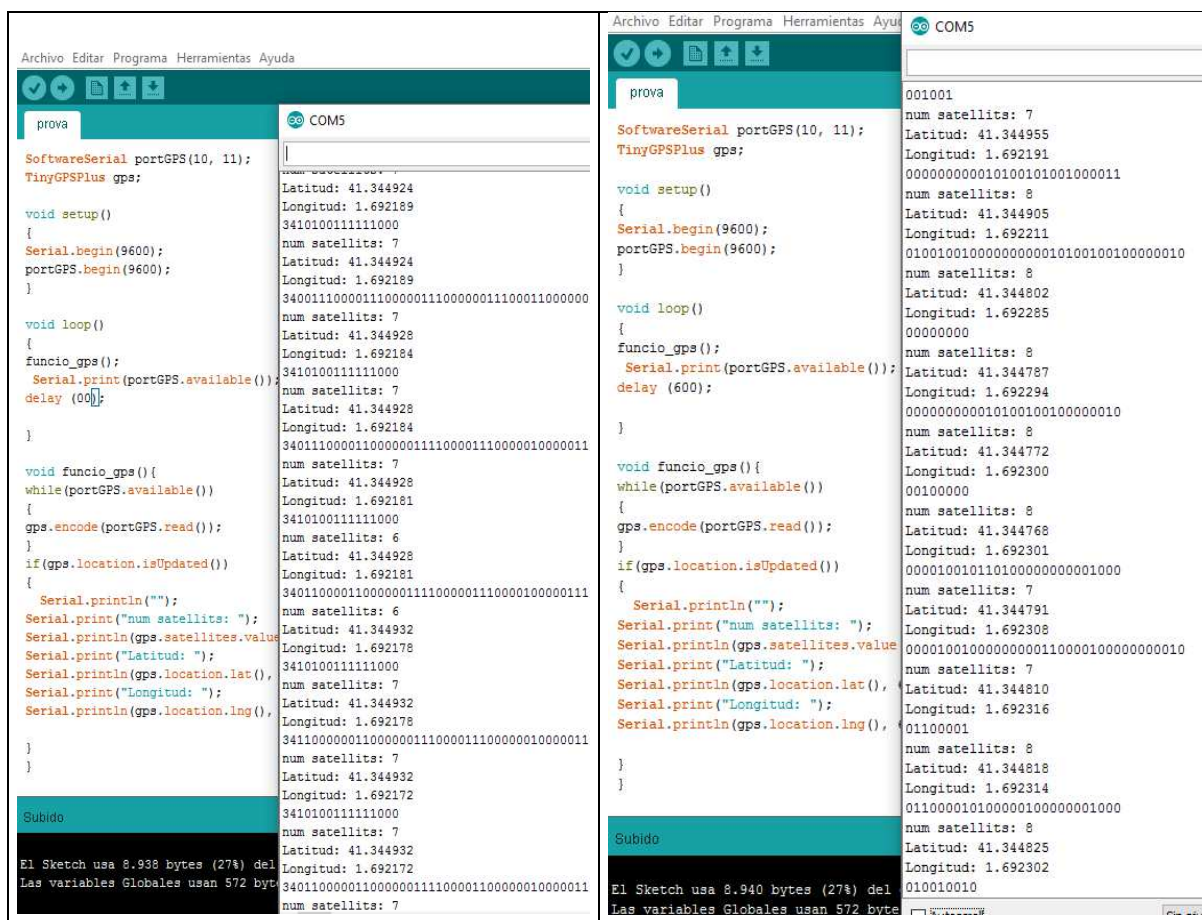


Figura 10.2. Imatges del monitor amb un retard de 0ms i un de 600ms.

La primera imatge es mostra sense retard. Cada nombre referent a la instrucció de disponibilitat s'actualitza molt ràpid, i podem observar que una gran majoria de temps val 0 i 1. Només el moment després d'escriure les dades val 34 anunciant que hi ha un flux de dades a llegir. El problema es dona en el segon cas que se li afegeix un retard de 600ms, aquí, cada cop que ha d'escriure un nombre ja tardarà 0.6s, i fa falta que doni la casualitat que aquest sigui d'almenys 30 perquè rebí dades que es puguin llegir. Val a dir que si el retard es fa més gran, de 0.7 segons, el programa s'estanca i no és capaç de donar més d'un cop informació. (Almenys amb els cinc minuts que s'ha estat esperant).

Tot i que hi poden haver solucions més òptimes, la que a mi se m'ha acudit ha estat la de obligar durant un temps el programa a mirar si hi havia dades a llegir. Si en comptes de fer-li passar un retard cada cop que busca si hi ha dades, li faig buscar durant cert temps amb un bucle, segurament acabi podent actualitzar la informació. També podria haver esperat fins que la instrucció de disponibilitat fos més gran de trenta per a poder continuar, però la primera m'ha donat bon resultat.

La instrucció per la que s'ha optat ha estat:

```
do { }  
while ( );
```

Amb la qual s'actualitza el valor d'una variable "e" 30.000 vegades i en aquest temps es dur a terme el bucle que escriurà nous valors de dades. El valor de la variable e, es retorna a zero a la funció `loop()` i es declara amb les llibreries com totes les variables de tipus *int*. S'ha de tenir en compte que aquest tipus de variable admet com a màxim valors de 32,767 pel que si superéssim aquest valor, hauríem de definir-la com a *unsigned int*, tot i així, no és necessari.

Amb aquest nou codi es poden posar retards tant alts com es vulgui que el programa sempre refrescarà les dades un moment més tard del valor del retard. A més s'han definit noves variables globals per als valors de nombre de satèl·lits, latitud i longitud que en un futur s'hauran de tractar.

10.2. Software del GPRS

Ara es programarà el mòdul SIM900 amb Arduino per a que sigui capaç d'enviar dades al servidor cada cert temps i no quan es premi la tecla q i s'envii al port sèrie tal i com vàrem fer a la última prova de la posada en marxa.

El servidor es troba ja operatiu i és capaç d'emmagatzemar dades, latitud i longitud, a la següent direcció:

<http://servidorvicencsuriol.herokuapp.com/messages/api/v1.0/arduino?lon=LON&lat=LAT>

On LAT i LON seran les coordenades que es volen enviar.

Al programa creat anteriorment s'entrava a l'índex d'un servidor també creat però sense estar programat amb una BBDD ni amb les operacions per a agafar dades que ja s'han dissenyat al apartat anterior. El primer a fer doncs, és comprovar que amb aquest nou link el programa envii coordenades de prova correctament.

Després d'algunes proves sense èxit, la direcció que s'ha de posar per a entrar al servidor com cal és aquesta.

```
char direccio []= "GET  
/messages/api/v1.0/arduino?lon=45776.876&lat=65898.564HTTP/1.1\r\nHost:  
holaeeee.herokuapp.com\r\nConnection:close\r\n\r\n";
```

Podem comprovar que primer s'ha de posar el directori del enllaç i després el Host on s'ha de connectar el mòdul. Cal tenir en compte que totes les barres, *r* i *n* són imprescindibles. Això és

perquè en C, `r\n` significa un enter. Si provem el codi que ja teníem amb aquesta nova direcció, podrem entrar al dataclip corresponent a la BBDD del servidor i observar com efectivament el codi ha penjat aquests valors. Exactament són els que tenen un id: 8, 9 al haver executat el codi dos vegades.

id	latitud	longitud
1	65	45
2	65.78	45.66
3	658	456
4	65898	45776
5	75458.6575	465466.56755
6	75458.6575	465466.56755
7	65898	45776
8	65898.564	45776.876
9	65898.564	45776.876

Figura 6.3. Captura de pantalla del dataclip de la Base de Dades.

L'objectiu és però que ho faci cada cert temps per ell sol i no pas enviant alguna cosa pel port sèrie. Per altra banda, quan hagi passat aquest cert temps, el que volem és que, d'alguna manera o altra, Arduino es resetegi juntament amb el mòdul SIM900 i torni a començar el codi de nou. Malauradament no existeix cap forma de fer un Reset per software d'un microprocessador d'aquests sense haver de pitjar el botó expressament [76] pel que s'haurà d'utilitzar a la força el `loop()` per a anar fent cicles.

El programa al `loop` finalment queda així:

```
void loop()
{
  iniciar();
  PeticioHttp();
  delay(10000);

  if (SIM900.available()) {
    Serial.write(SIM900.read());
  }
}
```

Abans fèiem el `iniciar()` al `Setup` però es vol que cada vegada que faci un cicle, es connecti de nou a la xarxa per si de cas l'antena 2G ha canviat o ja no és possible establir la connexió com a l'últim cicle. Per ara a més, s'hi ha afegit un retard de deu segons entre cicle i cicle. Després ja es millorarà tot el codi; Per ara, anem a fer que funcioni.

Un altre punt important és el fet d'encendre i apagar cada vegada el mòdul SIM900. Si s'ha de connectar cada cicle a la xarxa, aquest també s'ha de apagar i encendre de nou. Fins ara l'única manera de fer-ho era mitjançant el botó d'encesa, però s'hauria de fer per software.

L'única manera per aconseguir-ho és per hardware, que un pin simuli que ha pulsat el botó, és a dir, doni un pols alt un segon i torni a un estat baix. Per a fer-ho cal soldar un pont entre dos punts de la placa, el fabricant ho posa ja de manera senzilla per si cal prémer el botó per software com és el cas. El pin que cal donar el pols és el 9, i el codi que a mi m'ha funcionat ha estat aquest:

```
digitalWrite(9, HIGH);  
delay(1000);  
digitalWrite(9, LOW);  
delay(6000);
```

Després de declarar el Pin 9 com a sortida al Setup, s'encén 1 segon i s'apaga. Bé no té gaire misteri, Vout= 5V un segon, i després Vout= 0V. Després es dona 6 segons al mòdul per a què s'encengui. També es podria posar la comanda AT que el mòdul respon sempre amb un OK, si no respon, és que encara no s'ha encès. Però 6 segons són sempre suficients per a que s'encengui.

Per altra banda, també interessa apagar-lo quan hagi acabat d'enviar les dades, i tal i com es fa pitjant el botó, la manera de fer-ho és la mateixa que per encendra'l pel que el codi també serà el mateix.

Un altre fet important és que en el moment en què envia la petició GET, molts cops no era capaç de retornar el que el servidor li retornava tot i que a servidors més grans si que ho fa perfectament, potser és pel fet de ser el servidor de Heroku que de vegades no respon correctament. Però això no vol dir que la petició n' s'hagi fet correctament! I per tant les dades s'han emmagatzemat a la base de dades. Però per a què això no sigui un problema, cal que quan es posi el link del servidor no s'estigui esperant un OK per resposta, o es podria quedar allà i no tornar a fer un següent cicle.

Si s'ha fet tot això correctament, el programa serà capaç d'enviar dues dades fixes al servidor cada deu segons aproximadament i estem ja a un pas per a poder unir tot el codi Arduino i acabar l'aplicació en aquesta interfície.

10.2.1. Freqüència d'actualització de dades i el mode Sleep

Cada quan de temps s'ha d'enviar dades al servidor? Què ha de fer mentre es troba entre cicles?

Aquestes dues preguntes es troben lligades de la ma, perquè la tarifa de dades contractada no pot ser superada al mes i quan més ràpid envii coordenades, més ràpid es gastaran els Mb. Si parlem de un cicle de cada 10 segons, bé, ja podríem acabar el nostre programa. Però realment fa falta tanta freqüència?

Per altra banda, també es podria fer que si les coordenades noves amb les anteriors són molt poc diferents, que ni enviés les coordenades i així no gastaria, però si fos així mai es sabria si ha estat un error el que ha fet deixar d'enviar dades el mòdul, pel que per ara, la solució que sembla més òptima, és la de enviar dades cada cert temps.

Anem a veure el consum que porta el mòdul amb totes les proves que s'han fet fins ara:

La base de dades ha enregistrat 220 línies de coordenades, amb Longitud i Latitud enviades pel mòdul ja que la tercera columna, la de la data i hora, la posa el servidor mateix. Cal reduir però el nombre de files perquè aproximadament 100 d'elles s'han fet per a provar el servidor enviant les dades directament d'un buscador d'internet en un PC. Això fan doncs 120 línies.

Pel que fa a la tarifa de Simyo, la companyia et deix veure el consum mensual que portes pel que podem estimar quin ha estat el consum d'aquestes 120 línies i pensar el temps mínim que pot estar el programa entre cicle i cicle:



Figura 10.3. Pàgina de Simyo amb el consum del mes d'abril.

Podem comprovar que són 0.6 MB pel que els càlculs a fer serien aquests:

$$\frac{0.6 \text{ MB}}{120 \text{ Línies}} = 5 \times 10^{-3} \text{ MB/Línia} \quad (\text{Eq. 10.1})$$

$$\frac{100 \text{ MB}}{5 \times 10^{-3} \text{ MB/Línia}} = 20.000 \text{ Línies} \quad (\text{Eq. 10.2})$$

$$\frac{20.000 \text{ Línies}}{30 \text{ dies}} = 666.66 \text{ Línies/dia} \quad (\text{Eq. 10.3})$$

Si es volgués exprimir al màxim la tarifa de dades de Simyo, s'huria d'enviar dades al servidor 666 cops al dia. Però realment no volem passar-nos i estar ben segurs que no ho farem pel que aplicarem un 10% de marge pel que quedaria amb 600 vegades per dia. Això fan exactament:

$$\frac{\left[666.66 - 666.66 \times \frac{10}{100} \right] \text{ Línies/dia}}{24 \text{ h} \times 60 \text{ min}} = 0.417 \text{ Línies/min} \quad (\text{Eq. 10.4})$$

$$\frac{1 \text{ Línia}}{0.417 \text{ Línies/min}} = 2.4 \text{ min} \quad (\text{Eq. 10.5})$$

Doncs 2.4 min són els que hauríem d'esperar per a enviar les coordenades una i altra vegada. De manera subjectiva, em sorprèn que sigui cada tant poc temps i, per suposat me'n alegra, però s'haurà de fer alguna prova més endavant per a comprovar que efectivament els càlculs estimats són correctes.

Ara fa falta pensar que farà Arduino en aquests quasi dos minuts i mig, però cal recordar que el sistema és sense fils pel que l'ús d'una bateria és obligatori.

El sistema tarda més o menys 35 segons en realitzar tot el procés d'enviament de dades sense tenir en compte el sistema GPS. Podríem aproximar tot el procés inclòs el receptor GPS (que en bones circumstàncies tarda uns 10 segons en obtenir senyal GPS dels satèl·lits) uns 50 segons.

Aquest temps s'ha de restar al temps d'espera pel que el resultat queda de 1.6 minuts d'espera i 50 segons o 0.8 minuts de posada en marxa. Això es tradueix a que el 66% del temps total d'un cicle se'l passarà sense fer res, o el que és el mateix, només s'aprofitarà l'energia de 1/3 del temps d'ús del dispositiu. Si reduïem el consum i el fem estable en el temps, si aconseguíssim reduir a 0 la potència dissipada pel hardware en tot aquest temps, aconseguiríem teòricament, augmentar l'eficiència energètica un 200%.

Tots els càlculs de consum els farem en un apartat posterior que farà referència a l'alimentació del dispositiu i ara no entrarem amb més detalls, però ja s'entén que si hi hagués alguna manera d'apagar l'Arduino tot aquest temps i tornar-lo a encendre passats els 1.6 minuts, hi guanyaríem en energia, i això és clau en sistemes sense fils que depenen d'alimentació d'una bateria.

Malauradament no existeix cap mètode per a apagar un Arduino i tornar-lo a encendre per Software ja que quan hi ha alimentació, Arduino comença a executar el programa i es queda al loop fins que s'apaga. Però si que hi ha un mode hibernació per el microprocessador que desactiva algunes funcions i Pins del micro.

El difícil d'aquesta operació però, és com tornar a fer funcionar Arduino després d'entrar en aquest mode. La pàgina oficial explica que la única manera de fer-ho a part de fent un hard Reset és amb interrupcions.

Hi ha una manera més fàcil d'aplicar el Mode Sleep d'Arduino, una llibreria que s'anomena Low-Power-Master. Aquesta llibreria et permet hibernar Arduino i després el fa operatiu realitzant una interrupció ja programada. Tot i així no és massa difícil fer-ne una i a la pàgina oficial en dona un exemple [77].

Només fa falta una instrucció per a utilitzar la llibreria que controla el Sleep Mode d'Arduino, es presenta a continuació.

```
LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);
```

Aquesta funció posa a hibernar Arduino durant 8 segons. Hem comentat que el volem tenir hibernat 1.6 minuts almenys però el màxim temps que admet d'hibernació són 8 segons. Si volem allargar aquest temps, s'haurà d'utilitzar un bucle que repeteixi aquesta operació entre els quals el programa s'encendrà, augmentarà una variable per ràpidament tornar-se a hibernar 8 segons més. Desconec si aquesta metodologia pot portar algun efecte secundari en rendiment, durabilitat, etc. Del microprocessador però així hi guanyarem en eficiència energètica.

Si volem tenir-lo 1.6 minuts, és a dir 96 segons, s'ha de fer un bucle de la instrucció anterior de 12 cops. El codi quedaria així i s'afegiria després d'enviar dades i apagar el mòdul SIM900:

```
for (int i = 0 ; i < 12 ; i++)
    LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);
```

10.3. Unió i disseny d'un codi operatiu

Per separat ambdós codis funcionen bé. Anem doncs a unir-los per a donar lloc al codi final que ja podrà per fi, enviar coordenades GPS a un servidor d'internet.

L'objectiu és que, com que Arduino no es pot reiniciar, que fem tot el que ha de repetir cada cicle al loop, pel que en el setup no hi hauríem de tenir més que la inicialització dels ports sèrie. Per altra banda, s'ha d'ajuntar totes les llibreries, ports i variables en un sol codi.

El primer que farem serà inicialitzar el portGPS i fer totes les operacions que es feien per a captar senyal GPS. Al principi el codi donava problemes amb els dos ports per Software funcionant a la vegada pel que, per evitar qualsevol problema si és que és aquest el motiu, simplement quan acabi de fer totes les instruccions on s'involucra el receptor GPS, es tancarà el port amb la comanda:

```
portGPS.end();
```

I s'inicialitzarà la del SIM900. Cal recordar que els dos ports van a diferents velocitats de comunicació (9600 bauds el GPS i 19200 bauds el SIM900). Mentre el port GPS està obert, s'ha de cridar la funció GPS que per ara deixarem tal i com vam acabar en el seu apartat de disseny. Cal tenir en compte que el receptor tarda un temps en rebre senyal per primera vegada, uns 10 segons, pel que novament, per ara, posarem primer un retard i, per a assegurar una mica més, cridarem la funció GPS tres vegades seguides.

La funció GPS emmagatzemava les coordenades en dues variables float; Però resulta que si imprimim aquestes variables per pantalla veurem que només donen dos decimals i la resolució de dos decimals per a coordenades és excessivament pobre. Si no s'aconsegueix descobrir perquè són dos i com augmentar la xifra, s'haurà de pensar en una altra alternativa per a enviar aquestes dades.

Un cop acabat el codi GPS començarem amb el codi del GPRS. Realment el loop tindrà el mateix com el vam deixar en el apartat anterior, l'important ara és aconseguir posar les coordenades i nombres de prova a la URL que s'envia.

S'ha intentat ajuntar els String i Floats que componen el link i les coordenades de moltes maneres però Arduino és molt estricte i fa saltar errors per tot arreu. A més a més, en el cas d'aconseguir-ho, encara hauríem de pensar com augmentar els decimals del Float que, tot i que sembla culpa de com estan predefinits els Floats en el compilador, ni amb Double ni amb String milloren els decimals.

La solució a tots els problemes és, simplement, no ajuntar les cadenes de caràcters o dit pel seu nom, concatenar les variables. Realment quan li enviem al mòdul SIM900 la URL, li enviem uns caràcters que signifiquen enters i aplica per a saber que ja s'ha acabat d'enviar informació. Bé doncs, si enviem una informació rere l'altre pel port el resultat serà el mateix que concatenant correctament i després enviant-ho tot junt.

Per a resoldre el problema dels dos decimals (ja que el primer que es va provar va ser enviant els Floats) s'ha de seguir una metodologia similar a l'anterior; si en comptes d'enviar la variable Float enviem directament el que li ha arribat pel port sèrie del GPS i que sí que té tots els decimals que es vulguin, el programa funcionarà com es desitja.

Només falta tenir una cosa en compte. Si no concatenem, no sabrem com de llarg és el missatge a enviar mitjançant la instrucció:

```
sprintf(aux_str, "AT+CIPSEND=%d", strlen(direccio));
```

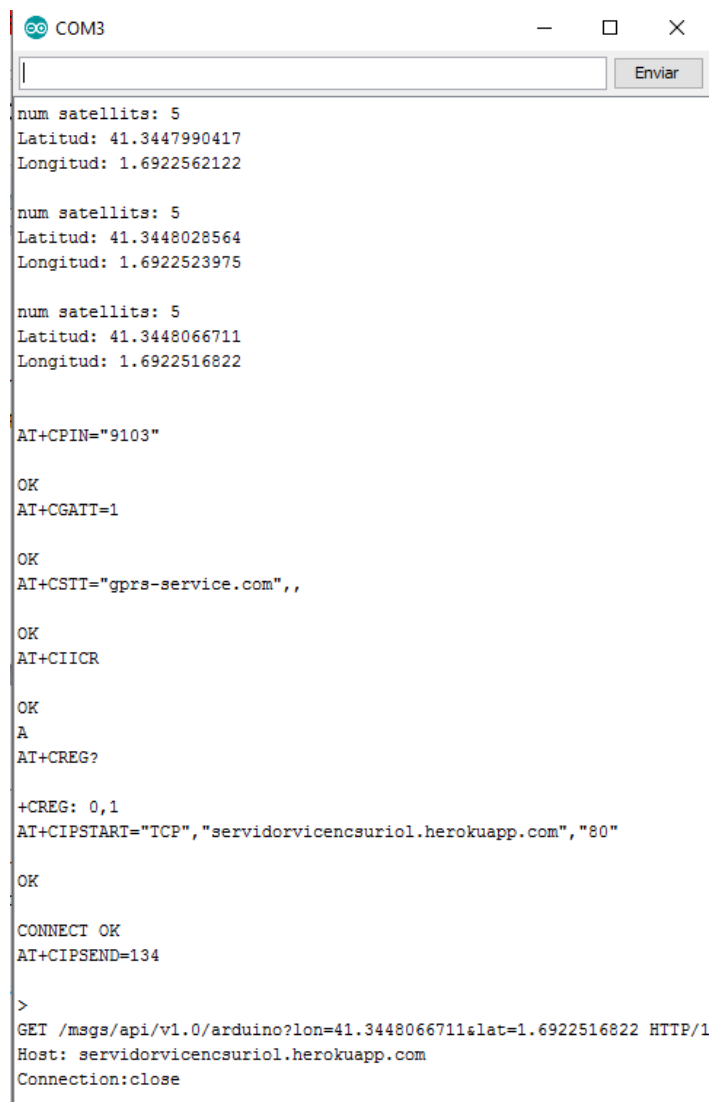
Que si recordem sprintf era una comanda per ajuntar a la variable aux_str la comanda AT que es mostra amb la longitud de caràcters de la direcció URL. Però ara la direcció ja no existeix. Bé doncs pel moment, s'ha buscat i introduït el nombre manualment pensant que els decimals de les

coordenades sempre seran els definits i el nombre enter tindrà dos valors de longitud i un de latitud a Catalunya. La longitud final és de 134 caràcters.

Amb tot això explicat, el codi final de la PeticioHttp queda desglossat per a que s'enviïn quatre línies en el CIPSEND. El codi es pot veure al Annex al apartat de codi per a Arduino.

Al final del codi, s'apaga el mòdul donant un pols d'un segon positiu al Pin 9 que hi ha el botó d'encès per software.

El resultat de tot el codi compilat i executat és el que es mostra a continuació:



```

COM3
num satellits: 5
Latitud: 41.3447990417
Longitud: 1.6922562122

num satellits: 5
Latitud: 41.3448028564
Longitud: 1.6922523975

num satellits: 5
Latitud: 41.3448066711
Longitud: 1.6922516822

AT+CPIN="9103"

OK
AT+CGATT=1

OK
AT+CSTT="gprs-service.com",,

OK
AT+CIICR

OK
A
AT+CREG?

+CREG: 0,1
AT+CIPSTART="TCP","servidorvicencsuriol.herokuapp.com","80"

OK

CONNECT OK
AT+CIPSEND=134

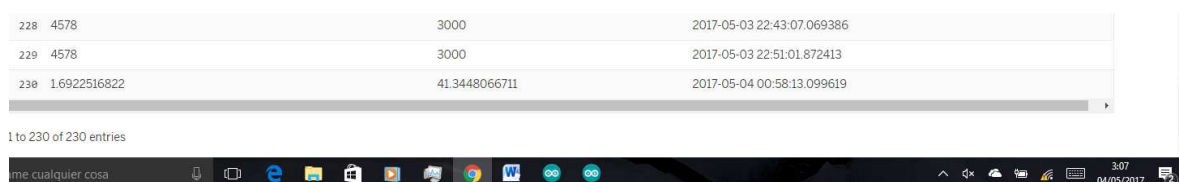
>
GET /msgs/api/v1.0/arduino?lon=41.3448066711&lat=1.6922516822 HTTP/1
Host: servidorvicencsuriol.herokuapp.com
Connection:close

```

Figura 10.4. Monitor sèrie amb l'execució del codi desenvolupat.

Es pot veure com primer es fa tres vegades la funció GPS, i després d'encendre el mòdul i esperar un temps, comença la funció iniciar. S'ha de dir que aquí s'han donat diferents errors tots relacionats amb la comanda AT+CIFSR. Al final es va descobrir que si el mòdul entrava en el while de si no estava connectat a la xarxa, que ho tornés a intentar, després la comanda AT+CIFSR mai tenia èxit i no es podia enviar res. Però si es connectava a la primera, la comanda no es mostrava pel port sèrie, només es veu una A però després tot funcionava perfectament. És molt estrany tenint en compte que tant en la primera prova com en el moment de crear el software pel mòdul tot funcionava bé així ni entrant al bucle.

Si seguim amb el monitor sèrie, podem veure com es connecta correctament al servidor i que la petició que hauria d'estar concatenada es simula correctament. El resultat final es pot observar en la base de dades del servidor:



228	4578	3000	2017-05-03 22:43:07.069386
229	4578	3000	2017-05-03 22:51:01.872413
230	1.6922516822	41.3448066711	2017-05-04 00:58:13.099619

1 to 230 of 230 entries

Figura 10.5. Pàgina amb les dades de la BD.

Podem observar que es tracta de l'última dada de la BD, els nombres són exactament els mateixos que hi ha a l'última presa de dades del codi del monitor sèrie anterior, i que la data a la qual s'ha pres és a les 00:58:13 del 04-05-2017 exactament la data i hora que marca el rellotge del ordinador, bé amb un retard de 9 min els quals s'ha fet la captura de pantalla i redactat la memòria. L'hora del servidor és l'hora internacional, i en aquestes dates, es troba dues hores per darrere de la d'Espanya, per això que marqui les 00:58 i no les 02:58 com realment són.

10.4. Robustesa del codi

Aquest apartat extraordinari s'ha afegit com a millora del codi, un cop provat i analitzat en profunditat s'han descobert diferents errors i s'ha decidit buscar-hi solució en aquest apartat.

10.4.1. Informació incorrecte del GPS

La primera vegada que s'encén el mòdul després de cert temps en funcionament, el GPS tarda un cert temps en agafar senyal GPS, de fet, és gairebé impossible agafar-ne dins d'edificis rodejats de blocs de pisos, soterranis o altres llocs que dificultin la senyal. Per això no té cap sentit donar-li 10

segons per a què es connecti, i ja donar per suposat que ho ha fet. S'ha de crear un bucle el qual no surti fins que el GPS funcioni, al cap i a la fi, tampoc serviria de gaire fer tot el loop sencer si les coordenades són tot zeros.

La idea era que fins que no hi hagi dades provinents del port, que es quedes a un bucle, però resulta que això no es pot fer perquè el receptor sí que n'envia de dades tot i no estar agafant NMEA dels satèl·lits. La solució s'ha trobat amb aquest codi que és una mica rebuscat però no se'm acut millor manera de fer-ho.

```
while(gps.location.isUpdated()==0){
  while(portGPS.available()){
    gps.encode(portGPS.read());
  }
  delay (100);
}
```

Afegida al principi del loop, mentre el port GPS no envii dades diferents a Arduino, que es quedi al bucle i no en surti. Per a fer això però, s'haurà de enregistrar totes les dades mentre aquestes s'enviïn pel port i un cop enviades totes, aleshores mirar si són diferents a les anteriors. Si no ho són, (es compleix el primer while == 0) i això ho mira cada 100 ms, no sortirà del bucle.

Ara ja en el moment que arribin dades diferents pel port, directament es posarà a fer la funció GPS, que realment no és més que una extensió d'aquest mateix codi de confirmació i segurament es pogués evitar. Tot i així tampoc comporta cap impediment i de vegades les primeres o segones dades que rep el receptor no són del tot acurades.

10.4.2. Reiniciar per software el receptor GPS

Un altre punt que s'ha volgut millorar és el fet de tenir el GPS sempre encès. Això és un consum, per més mínim que sigui segueix essent consum. La solució a poder parar i reiniciar el receptor GPS és alimentant-lo per un port de sortida d'Arduino. El problema és que quan el mòdul es troba ancorat a els satèl·lits, el seu consum és baix però quan es troba en la cerca de senyal el consum és més elevat dels mA que pot proporcionar qualsevol port d'Arduino pel que mai s'acaba connectant i enviant les coordenades al microprocessador.

És veritat que es poden buscar alternatives per a què això funcioni, però justament quan consumeix més és quan està buscant senyal dels satèl·lits i si a cada cicle el fem passar per aquest procés, és probable que més que estalviar-nos consum, l'augmentem. Per altra banda, el mòdul mai ha donat cap error com per tenir la necessitat de reiniciar-lo, pel que per aquesta banda tampoc faria falta una alimentació controlada. És per aquests dos motius que s'ha decidit no continuar amb fer funcionar aquesta nova implementació.

10.4.3. Nombre de satèl·lits

El nombre de satèl·lits els quals el receptor ha agafat les coordenades pot servir per a especificar la qualitat en la precisió de les coordenades mostrades, i ara mateix, tot i que el microprocessador rep aquesta informació, no la tracta ni l'envia. I vista la similitud de la tasca amb les coordenades, es pot afegir aquesta informació per a ampliar el projecte i afegir-li aquesta nova funcionalitat.

Pel que fa al microprocessador, el que és més important és saber la longitud de la nova línia que s'escriurà per fer la comanda de AT+CIPSEND que envia al servidor la sol·licitud. Per a poder rebre correctament la nova variable, el servidor ho haurà de contemplar i és això el que es farà primer de tot.

```
numsatellites = request.args.get('nums')
```

Aquesta línia demanarà al GET la nova variable que s'haurà d'introduir amb el nom de noms i l'afegirà a la pròpia variable numsatellites que serà tractada a la base de dades amb aquesta nova línia.

```
numsatellites = db.Column(db.Integer)
```

No fa falta que sigui un String com les altres dues variables (que de fet podrien ser Floats) al cap i a la fi, el nombre de satèl·lits només podrà valdre valors enters.

Amb totes les modificacions fetes al servidor (que són més però són molt intuïtives), podem programar el microprocessador per a què envii el nombre de satèl·lits. Com que el factor de la longitud del GET és important i amb un sol caràcter més el codi deixaria de funcionar, és important que el nombre de satèl·lits sempre tingui la mateixa longitud. Fins ara, no s'ha vist més de 10 satèl·lits amb una mitjana de 6 pel que limitarem que si són més de 10, la variable que enviarem sigui 9. No importarà aquest detall alhora de dir la qualitat de les coordenades.

```
nsat = gps.satellites.value();  
if (nsat>9){  
    nsat=9;  
}
```

I per a enviar-ho caldrà posar mentre estem afegint les coordenades al port serial:

```
SIM900.print("&nums=");  
Serial.print("&nums=");  
delay(50);  
  
SIM900.print(nsat);  
Serial.print(nsat);  
delay(50);
```

Per últim, la nova longitud de la cadena resulta amb 141 caràcters.

Els resultats en la comprovació de la base de dades del servidor són els desitjables.

10.4.4. Coordenades negatives

Per a què el codi d'Arduino sigui definitivament robust, queda un punt molt important amb el qual no es podria considerar acabat sense ell, i és el fer la longitud de caràcters del GET una variable.

Fins ara no hi ha hagut problema amb l'enviament de dades perquè sempre tenien el mateix nombre de caràcters, 141 en concret, però si es mou cap al centre de la península, on hi ha una longitud negativa i per tant, un caràcter addicional, el codi no funciona. Tampoc funcionaria si anéssim a Itàlia per exemple, on la longitud supera les desenes.

El més òptim seria concatenar i després utilitzar una simple funció de length, però ja s'ha dit que aquesta idea no funciona, s'ha provat i no dona resultat.

La següent alternativa que se'm acudeix és la de variar una variable que si la longitud és major de 10, augmenti una variable en un, si la latitud és negativa, l'augmenti en un altre i així etc. Si es busca un mapa de coordenades, la latitud pot prendre valors del -85 al 85 però pel que fa la longitud, aquesta pot valdre valors superiors als 100.

Bé, són molts casos i el codi tindria moltes línies amb condicionals per una simple variable. Una altre idea, més o menys una mescla entre les dues alternatives se'm va ocórrer abans de programar tot aquest codi: Declarar longitud i latitud com a strings, demanar els caràcters amb length() de cadascun d'ells i sumar-ho a una base. Així, si el nombre era negatiu o superior a 10 o 100, length() ho contemplaria. El codi ha quedat així.

```
latitud = gps.location.lat();  
longitud = gps.location.lng();  
lati = latitud.length();  
lon = longitud.length();  
n = 132+lati+lon;
```


11. Creació de l'aplicació mòbil

Aquest apartat tractarà tot el procés de creació de l'aplicació mòbil, des del disseny inicial fins al resultat final. Es pot trobar tot el codi al Annex ja que no és l'objectiu comentar-lo tot en aquest apartat bàsicament perquè hi ha molt codi repetitiu i molts elements de poc interès un cop ja en saps el seu funcionament. Tot i així, sí que es comentaran totes les coses noves i s'afegirà el codi necessari per a entendre-ho i fer-ho més visual.

11.1. Què ha de fer l'aplicació? Objectius inicials.

Què és exactament el que ha de fer l'aplicació abans de posar-se a treballar. Com a idea general, l'app ha d'entrar a un servidor al qual hi ha emmagatzemada la informació, cada cop que s'obri aquesta i/o quan ho demani l'usuari, per a agafar totes les dades més actuals i mostrar-les per pantalla de manera que aquest pugui saber on es troba el prototip i, per tant, el seu cotxe, motxilla, un delinqüent, l'avi...

Anem a definir exactament els objectius de l'aplicació:

- Com a objectiu previ, l'aplicació haurà de ser capaç d'entrar a internet, connectar-se a un servidor i ser capaç de diferenciar les dades que hi ha allà per a poder agafar les més actuals o les que interessin en el moment. També haurà de tenir una interfície còmode i agradable per a poder interactuar amb ella fàcilment.
- L'objectiu més important i essencial és el poder mostrar per pantalla un mapa de Google on mostri la posició exacta de l'últim moment que el prototip ha enviat informació al servidor. Això es podria fer cada cop que l'aplicació s'encengui i cada cop que l'usuari polsi un botó de comprovació o refresc.
- L'app ha de ser capaç de realitzar un seguit de càlculs amb les dades rebudes com són la velocitat mitjana en funció del temps, mostrar la direcció a la qual es dirigeix actualment, algun càlcul estadístic com hores en moviment i parat els últims dies, entre altres.
- Afegir un conjunt d'alarmes on l'aplicació determini qualsevol anomalia en les dades rebudes del servidor ja sigui, per exemple, que el dispositiu hardware porta massa temps sense enviar dades, les dades siguin incompatibles ja sigui per el canvi sobtat de posició entre intervals de dades, o dades incorrectes per un error d'enviament del mòdul o de comunicació, etc.
- També es pot afegir la possibilitat de crear una zona prohibida o restringida on el dispositiu no podria trobar-se i si en fos el cas, l'aplicació donaria una alarma al usuari, o que si s'allunyés més de x metres de radi d'un punt, saltés.

Però cal tenir en compte que l'enviament de dades per la xarxa de comunicacions no és gratuït. En cap cas es podria tenir que el mòdul enviés dades cada segon o minut com ja s'ha vist. També es podria definir que enviés dades si el mòdul complís certes condicions com excés de velocitat, posició restringida... però hauria d'estar preestablert o en tot cas, que l'app pogués enviar dades per al canvi de variables com les explicades per a què el dispositiu les pugui agafar i modificar-les. La idea és bona però augmenta la dificultat ja que tant el mòdul com la aplicació haurien de poder agafar i transmetre del servidor.

- Un altre bon punt per on enfocar-se és amb la interacció entre la posició de l'usuari i del aparell. Si aconseguixo tenir accés al GPS del mòbil, podria donar dades com la distància entre el mòbil i el últim punt rebut pel dispositiu. Això facilitaria molt el rastreig del dispositiu.
- Per últim, aquesta aplicació es pot ampliar fàcilment, per exemple càlculs de simulació per a intentar predir on es trobaria el dispositiu des de l'últim cop que ha enregistrat dades tenint aquestes en compte. Ja es veurà en funció al temps que es tingui per a poder ampliar el projecte.

Igual que alhora de programar Arduino, s'ha fet un diagrama de flux per a poder entendre més bé quin esquema seguirà el projecte.

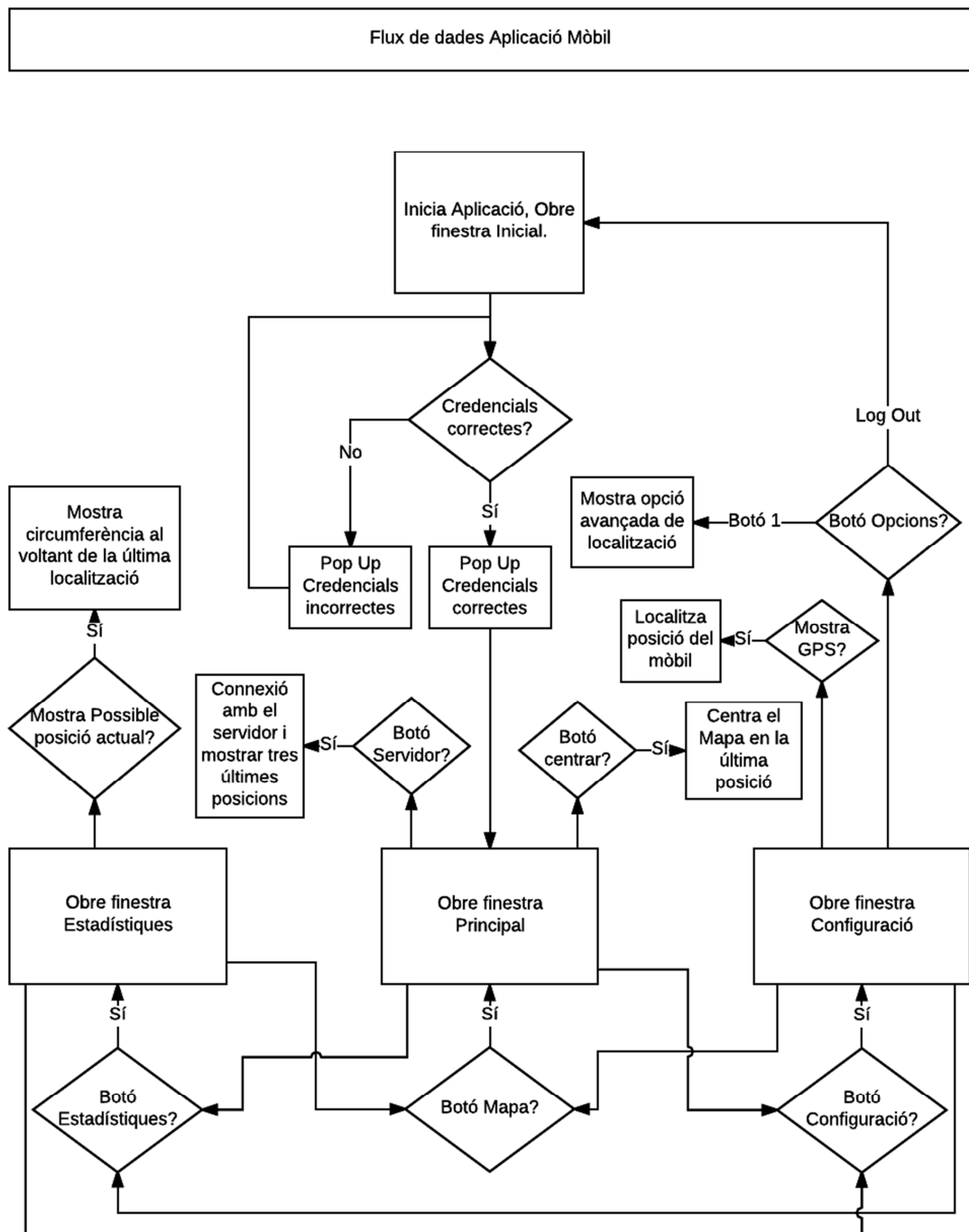


Figura 11.1. Diagrama de flux que es seguirà per a programar el codi de l'aplicació.

11.2. Comparativa amb una aplicació real

A continuació mostro una aplicació real similar al objectiu, pensant però que no hi haurà cap igual al mercat perquè la meua estarà enfocada a “trackejar” un dispositiu únic per a satisfer les necessitats

d'un treball final de grau. També n'adjuntaré diverses fotografies per a fer més visual la meua explicació.

"GEO Tracker- GPS tracker" de *Ilya Bogdanovich*.

Aquesta aplicació tracta sobre rutes, l'app enregistra cada cert temps la posició del Smartphone i la guarda a una base de dades per després poder-la mostrar en un mapa.

La pantalla principal és un propi mapa de Google que mostra la ruta que s'està fent al moment juntament amb un seguit de botons per a començar a enregistrar, acabar la ruta, ampliar i reduir el mapa, centrar a la posició actual. També té altres menús per a veure les rutes guardades, per a guardar punts importants al mapa, i un per a veure un conjunt d'estadístiques: longitud de la ruta, velocitat màxima i mitja, duració del recorregut, etc.

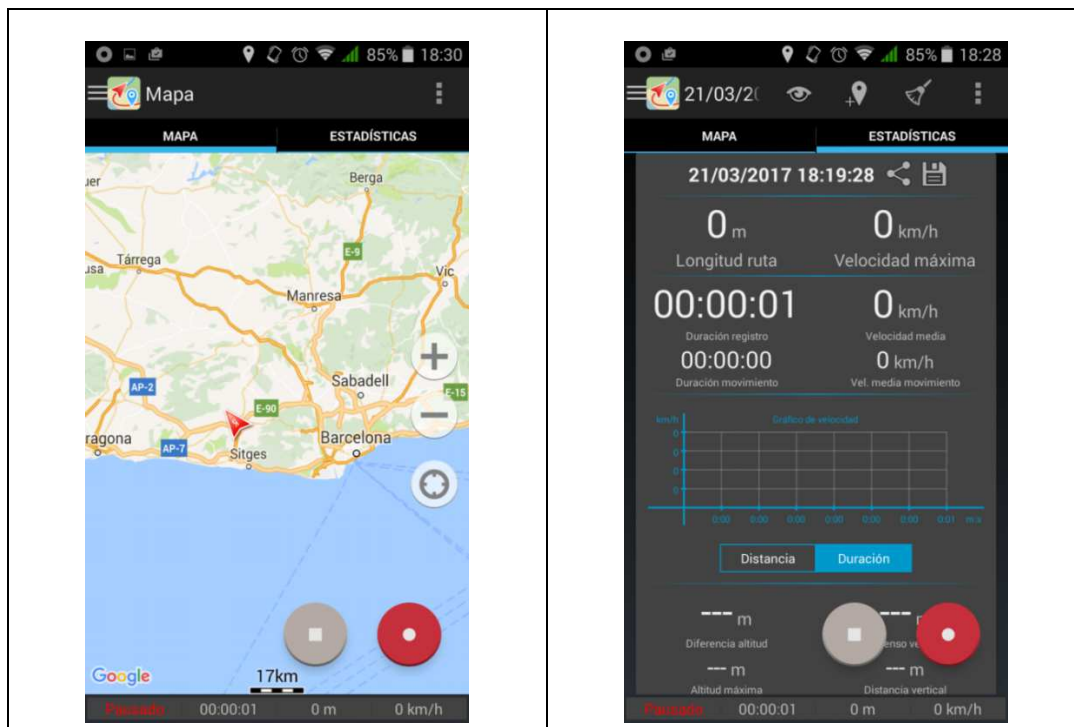


Figura 11.2. Pantalla principal i d'estadístiques de l'aplicació referent (Font: Ilya Bogdanovich).

Com ja s'ha explicat, l'app de manera general s'assemblarà a aquesta exposada però realment el nucli de programació no tindrà res a veure; El entrar a internet per a agafar les dades d'un servidor, etc. No té res a veure, i tot i que potser existeixi alguna app com la que es vol fer, el Play Store de Google et mostra les més descarregades i destacades, i tot i que hi ha moltes aplicacions referents a

GPS, o GPS Tracker, totes venen a donar una idea similar a la exposada o simplement compleixen amb la funció de Google Maps amb una altre interfície.

11.3. Disseny inicial

Comencem doncs pel principi, tota bona aplicació ha sorgit d'una idea i un prototip senzill al qual se li han anat afegint més elements i complexitat per acabar realitzant la funció a la qual es volia arribar. Toca doncs pensar com seran les pantalles de la app i tot el que hauran de tenir.

11.3.1. Esquema bàsic i interaccions entre els elements

Sense haver de tocar encara el programa Qt Creator, es pensarà i es dibuixarà de manera molt aproximada com serà l'app per a tenir una idea més visual del que es necessita.

Es pot primer de tot afegir una pantalla per a que l'usuari sàpiga que ha entrat a aquesta aplicació, pot contenir un logotip personalitzat i potser una barra de càrrega de l'aplicació mentre aquesta s'obre. El que haurà de fer en primera instància qualsevol persona que vulgui saber on es troba el dispositiu és identificar-se, pel que la segona pantalla serà d'identificació, amb la recepció de un nom d'usuari i una contrasenya obligatoris per a poder passar a la següent pantalla que es podria dir-li ja la principal. A més, si ho pensem, no té cap sentit poder registrar-se a la pàgina, només existeix un dispositiu i només un grup reduït i conegut de persones hauria de tenir permesa l'entrada a l'aplicació. Això facilita una part del treball, ja que l'app no necessita connectar-se a un servidor per a poder comprovar dades dels usuaris que vulguin entrar i només posant una clau correcta que tingui el propi codi programada podran entrar. A tot això se li hauria d'afegir una pantalla o un pop up que indiqui si són o no correctes les dades introduïdes, un títol, un botó d'informació... ja s'anirà veient més endavant.

La pantalla principal haurà de tenir un mapa de Google que per ara no s'entrarà a com posar-lo, però haurà de ser capaç d'augmentar i disminuir l'àrea que avarca com ho fa un mapa de Google normal i potser també un que centri la posició actual del dispositiu. A més hauria de tenir una barra d'eines per a que es pogués anar canviant entre menús. També seria interessant contemplar algun botó que fes entrar a l'aplicació a un servidor per a que agafés la última coordenada penjada si així ho vol l'usuari.

Una altra pantalla ja secundària però important, seria la d'estadístiques, que permetés veure un seguit d'informació sobre el dispositiu com per exemple velocitat mitjana en les últimes 24h, distància recorreguda entre d'altres.

L'aplicació també pot disposar d'una entrada de dades manual en el cas que es vulgui fer així, és a dir, si es sap la posició del dispositiu actual i es vol anotar sense que l'aplicació es connecti al servidor ja sigui per falta de internet, o per una mala gestió del mòdul 2g, es podrà introduir amb un menú manual. Aquest haurà de comptar amb tres espais per a introduir les coordenades i l'hora i un botó per a penjar les dades.

També de manera ampliable es poden afegir diferents menús d'opcions com ja s'han explicat a l'apartat anterior sobre què ha de fer l'aplicació.

Per a posar un exemple i tenir la informació més visual, s'ha fet un dibuix esquemàtic de les finestres sobre paper. També es podria haver fet a ordinador però no s'ha trobat necessari ja que per fer-ho amb ordinador, millor fer-ho directament al programa Qt que és el que es farà en el següent apartat. Tot i així s'inclourà el dibuix per a ajudar a entendre la filosofia que adoptarà l'aplicació.

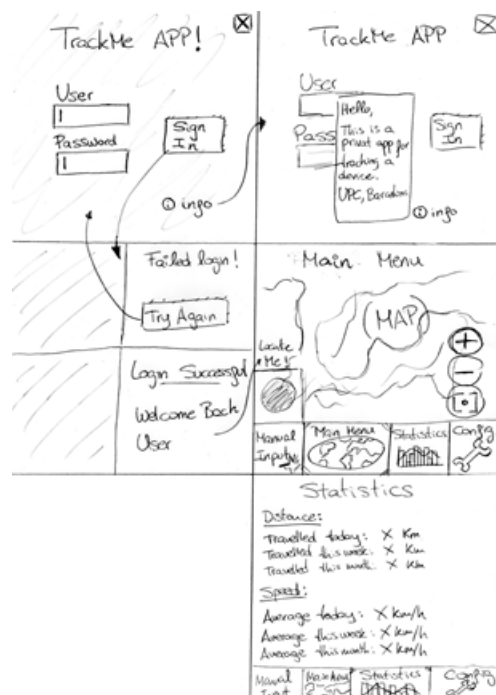


Figura 11.3. Dibuix explicatiu de les pantalles de l'aplicació a dissenyar.

11.3.2. Disseny de les finestres de l'aplicació

Per programar l'aplicació, s'anirà seguint l'esquema del dibuix del apartat anterior i s'anirà explicant tots els conceptes utilitzats. També s'ha de mencionar que cal un codi que reaccioni al canvi de pantalles i aquest serà la columna a la qual es dividiran la resta de aplicacions. Vindria a ser el loop i les finestres les funcions que es criden quan es compleixen certes condicions.

11.3.2.1. El codi base o Main.

Per a poder alternar pantalles dins de l'aplicació s'utilitza una funció específica (de fet hi ha diverses maneres de fer-ho). Aquesta s'anomena així:

```
StackView {}
```

Dins d'aquesta funció, es declaren components que aniran sortint com a principals quan se'ls hi ordeni. Per a entendre el funcionament d'aquest codi, el que es fa internament és sobreposar un component d'un altre quan se li ordena, i després l'amaga per a tornar a mostrar el primer component. StackView ha de ocupar tota la pantalla i se li ha de declarar el primer component que es visualitzarà per defecte.

Abans de mostrar el codi, explicar que el dibuix anterior era un concepte d'art i al final s'ha decidit no introduir la finestra de entrada manual perquè totes les funcions del servidor són prou competents com per evitar la necessitat de afegir un sistema per a introduir les coordenades manualment. Aquesta finestra doncs, s'ha eliminat i substituït per una de configuració que ja es veurà més endavant quins elements contindrà.

El codi del arxiu main doncs començarà tenint un aspecte com aquest:

```
Window {  
  id:root  
  visible: true  
  width: 555  
  height: 1000  
  StackView {  
    id: miStackView  
    width: parent.width  
    height: parent.height  
    initialItem: finestrainicial  
  }  
}
```

Tots els components afegits al StackView són arxius .qml com el propi main. A més per ara es trobaran tots menys el inicial en invisible perquè de vegades hi ha errors entre ells.

Si es veu el dibuix, la manera per canviar de pàgines un cop s'ha superat la pantalla inicial, és amb tres botons fixes a baix de la pantalla. Bé doncs aquests botons no es troben a cap finestra en concret, pel que és una bona idea posar-los en aquest arxiu també. Si els poséssim en algun altre com en la pantalla de estadístiques, quan canviéssim de finestra els botons desapareixerien i això no interessa.

Els botons no haurien de ser rectangles, més que res per estètica. S'han agafat imatges sense llicència d'internet per a fer-los. A més el icona que hi haurà a sobre de cada botó també són imatges retocades per a combinar bé amb el fons negre del botó i se'ls hi ha extret el fons per a que quedi transparent. Primer s'ensenyarà el resultat i després com s'han programat:



Figura 11.4. Disseny dels botons.

Ja es veu que cada botó conté una imatge negra, una d'icona i un text a més que quan acaba un comença l'altre i el central es troba ancorat al centre de la finestra. La programació d'aquests elements és aquesta:

```
Image {
  id: botost
  source: "imgs/blank-black-button-md.png"
  width: root.width/3.6
  height: root.height/12
  x: botomap.x- botost.width
  y: root.height-botost.height*1.2
  visible: false
  MouseArea{}
```

La primera imatge correspon al requadre negre, té una amplada i una altura que varien en funció de les dimensions de la pantalla, és a dir, del mòbil; i la seva posició és una mica rebuscada, haurà de estar una mica més pujada del final de la pantalla, d'aquí aquesta y, se li resta 1.2 vegades l'altura de la pròpia imatge de l'altura de la pantalla, i les x, ja s'ha dit que ha d'estar just abans que comenci la següent, que al haver-hi tres botons estarà centrat a la pantalla, pel que s'ha de fer referència a la posició d'aquest rectangle central que encara no s'ha vist i restar-li l'amplada d'aquest botó.

```
Image {
  id: imatgestatistic
  source: "imgs/stats.png"
  width: root.width/8
  height: parent.width/2.2
  x: parent.width/2 - imatgestatistic.width/2
  y: root.height/24 - imatgestatistic.height/2 -5
}
```

La segona imatge correspon al icona d'estadístiques. L'amplada i l'alçada han d'estar relacionades o amb les dimensions del pare (imatge del rectangle) o de la pantalla, i han de ser menors a les del rectangle. La seva posició li passa el mateix que en les dimensions però s'han de tenir dues coses en

compte: La imatge es troba dins de la primera imatge pel que el 0,0 és el 0,0 de la primera imatge i no pas de la pantalla, i quan es situa, s'està situant el seu punt 0,0 no el seu punt central pel que si es vol centrar s'haurà de restar la meitat de la seva amplada i altura.

```
Text {  
  id: statistext  
  text: qstr("Statistics")  
  color: "white"  
  font.family: "Times New Roman"  
  font.pointSize: botost.height/5  
  y: botost.height -20  
  x: botost.width/2 - 34  
}  
} //boto estadístiques
```

Per últim el text es troba de color blanc, es declara la font i com de gran serà aquesta, en aquest cas variarà en funció de l'altura de el rectangle negre, i la seva posició serà per ara una que no variï però que la centri en funció de com de llarga és la paraula. Si es fa el mateix amb els següents botons variant la posició d'ells la resta de codi queda així. Deixo l'estudi d'ell per a qui li interressi al ser molt semblant a el exposat en el primer.

Amb tot això ja estaria programat el primer arxiu que cridarà els altres depenent del botó que es premi. Malauradament no es pot provar el codi fins que no s'hagi programat almenys una altra finestra.

11.3.2.2. Pantalla Inicial

A partir d'ara ja no seran noves finestres el que es creïn, perquè si fossin finestres cada vegada s'obriria una de nova i realment es vol que un component passi a sobre de l'altre però a la mateixa finestra. Per tant doncs es crearan Items.

El primer que es posarà a mode de fons serà una imatge d'un mapa mundi. Per a que no prengui molt protagonisme, s'ha posat transparència al 25%.

```
Rectangle {  
  id: fons  
  color: "#00000000"  
  Image {  
    x: -291  
    y: 0  
    height: root.height  
    sourceSize.width: 0  
    z: 3  
    source: "imgs/world.jpg"  
    opacity: 0.75  
  }  
}
```

Root és com s'ha definit la finestra al arxiu Main i per això surt en blau. No té amplada per a no deformar la imatge, ja que és molt llarga. A més la x és negativa per a desplaçar la imatge cap a l'esquerra i que així quedi més ben centrada.

A continuació s'afegeix un títol per a la aplicació. No quedaria bé posar un amb lletres normals pel que s'ha buscat unes lletres més elaborades i s'ha fet una imatge amb fons transparent per a afegir-la a la pantalla. Cal dir que el títol no s'ha pensat gaire temps i potser és temporal.

La posició s'ha de centrar en funció sempre de la pantalla, i les dimensions són les adequades després d'una mica de tanteig.

Passem ara al requadre amb el Usuari. En ell s'ha de crear una entrada de text, un text per a referir-nos a Usuaris i el rectangle que farà de base de la entrada de text.

El rectangle es troba ubicat just al centre de la pantalla i a una tercera part d'altura d'aquesta, i les seves dimensions són mitja amplada de la pantalla i una vintena part d'altura. El text és més interessant perquè es troba a una altura negativa. Això és perquè el seu 0,0 coincideix amb el del pare, el rectangle pel que si es vol posar a sobre d'ell, l'altura haurà de ser negativa, en concret val la pròpia altura del rectangle. El text Input et permet introduir text, es troba centrat en el rectangle i té gairebé les seves dimensions. Té la propietat `maxLength`, que et permet dir la màxima longitud de caràcters que es poden introduir. En aquest cas es troba a 13 caràcters.

```
TextInput {
  id: user
  width: parent.width-15
  height: parent.height-20
  font.family: "Courier"
  maxLength: 13
  anchors.centerIn: parent
  focus: true
  font.pixelSize: 30
}
```

Pel que fa a la part de la contrasenya, l'esquema a seguir serà molt similar, l'únic que canvia són la posició, una mica més a baix, el text a mostrar i la propietat del `TextInput`, `echoMode`; Aquesta propietat tal i com està definit, permet amagar els caràcters escrits substituint-los per cercles negres com ho fan moltes aplicacions al introduir una contrasenya.

El següent a programar és el botó per entrar i el botó d'informació que tenen una programació molt semblant. No considero necessari comentar el codi en aquests dos vists ja els tutorials i tot l'anterior.

Per últim, s'ha creat un logotip que representa l'abast de tot el projecte en sí. Es col·locarà a la cantonada inferior dreta i ha quedat així:

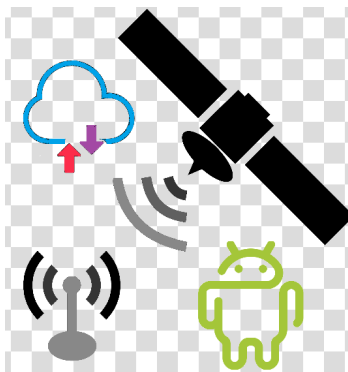


Figura 11.5. Logotip de l'aplicació.

El codi no té massa misteri tampoc, una imatge amb posició i dimensions. El resultat ja es pot executar i veure com ha quedat.



Figura 11.6. Pantalla inicial de l'aplicació.

La primera pantalla encara no està, falta dues semi finestres que confirmen si s'ha registrat o no correctament.

Com que realment aquestes dues finestres són pop ups, finestres que apareixen sobre la principal igual que la del botó d'informació, aquestes tres es comentaran ara. Per ara seran imatges d'un fons de color verd i vermell, juntament amb un icona que també és una imatge i text. Pel que fa al

requadre de que no s'ha introduït el usuari correctament, haurà de tenir un botó per a tornar-ho a intentar. El codi és més llarg però el més interessant és el Label de a continuació.

```
Label {
    text: "Maybe you should consider if this is the app you
    are looking for ! "
    anchors.margins: 20
    anchors.left: parent.left
    anchors.right: parent.right
    font.pointSize: parent.width/22
    y: parent.width/3
    wrapMode: Label.Wrap
}
```

El wrapMode s'utilitza per a que quan el text sigui llarg i sobrepassi els límits del pare, que segueixi a la següent línia.

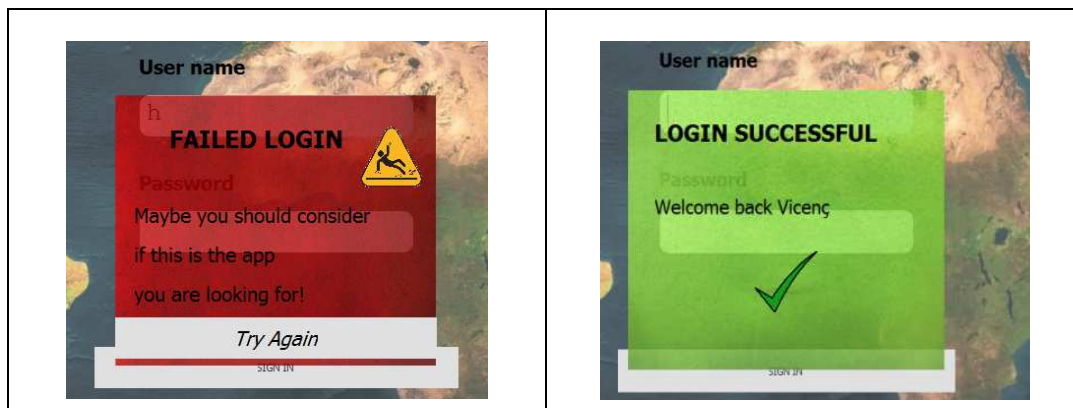


Figura 11.7. Pop ups d'entrada.

Recordar que no s'ha mostrat tot el codi perquè no és pas la idea. Tot i així el codi complert es troba al Annex.

11.3.2.3. Pantalla principal

La pantalla principal ha de contenir el mapa per a visualitzar la posició del servidor. Per a poder posar un mapa amb qml (el llenguatge de programació de Qt) m'he basat amb un vídeo que va publicar un dels dissenyadors de Qt. Es pot seguir en el següent enllaç [78].

El codi funcional del mapa és aquest:

```
Map{
    id: map
    anchors.fill: parent
    plugin: Plugin{name: plugin/"osm" "here" }
```

```
PluginParameter { name: "here.app_id"; value:
"0rXNQ9WuJAUsGhd6jxAk" }
PluginParameter { name: "here.token";
value: "PcZTaHkMDQhndMfEIGpBOQ" }}
center: QtPositioning.coordinate(41.4136865,2.2221190000000206)
zoomLevel: 15}
```

On plugin representa el servei que ofereix un mapa que vols fer servir, (en aquest cas es troba en la variable plugin que val osm, però també pot valdre here, un mapa que necessita registre per a utilitzar la seva API i s'ha de declarar dos paràmetres que et donen quan et registres). OSM fa referència a Open Street Maps, i Here és l'altre companyia que ofereix un servei de mapes per a aplicacions mòbils, però al contrari de OSM, no és gratuït al llarg del temps, donen 3 mesos per a que provis el servei. El mapa es centra a les coordenades de la EEBE i fa un Zoom de 15 nivells.

El resultat dona un mapa interactiu que permet ampliar-se i moure's per ell en llibertat.

En el mapa però, caldrà afegir-hi marcadors que no seran simples imatges, caldrà que estiguin ancorades al mapa, que es moguin amb ell sense fer-se petites al disminuir el mapa.

Això es fa amb MapQuickItem i ha d'estar dins el mapa. Es posarà un d'exemple tot i que s'ha afegit uns quants Items.

```
MapQuickItem{
    id:quickitem
    coordinate: QtPositioning.coordinate(latitud0,longitud0)
    sourceItem: Image {
        id: marker
        source: "imgs/redmarker.png"
        width: root.width/11
        height: width*1.6
        visible: false
        MouseArea{
```

El QuickItem conté la posició a la qual va ancorat, en aquest cas dues variables a les que s'afegirà més endavant la posició del servidor, i una imatge que serà invisible de moment i tindrà una Mouse Area per a poder interaccionar amb la icona per a què aparegui un requadre amb la data que s'ha pres la captura.

Per a fer que la imatge estigui justament centrada a les coordenades, caldrà afegir aquestes dues línies de codi:

```
anchorPoint.x: marker.width/2
anchorPoint.y: marker.height
```

Deixant de banda el mapa, aquesta finestra també tindrà un botó que entri al servidor i un que centri la pantalla a les coordenades que es trobi la posició del servidor.

La imatge de centrar el mapa tindrà una Mouse Area així:

```
MouseArea{
    anchors.fill: parent
    onClicked: {
        map.center= QtPositioning.coordinate(latitud0,longitud0)
        map.zoomLevel = 14
    }
}
```

Que centrarà el mapa creat abans a les coordenades desitjades i deixarà el zoom a 14.

El botó per a connectar-se al servidor és ja més detallista:

```
Button {
    id: trackme
    width: botost.width
    height: width
    x: botost.x
    y: centrarmapa.y
    text: qsTr("Track Me")
    onClicked: {

        request("https://servidorvicencsuriol.herokuapp.com/messages/api/v1.0/a
pp",procesaResposta, "GET", "")
        actualitza.running=true}
    }
```

En el moment que es polsi, farà un GET al servidor que s'ha creat que conté les coordenades enviades per Arduino. Request i CreaCallback, són funcions que s'han extret de una aplicació que es connectava a internet pel que no s'han dissenyat personalment. Es poden trobar al Annex, però el que sí s'ha fet ha estat procesaResposta, que com vol dir el nom, farà totes les funcions necessàries amb el resultat de request i es programarà més endavant ja que per ara s'està creant simplement el disseny extern de l'aplicació.

El disseny de la finestra final és el següent:

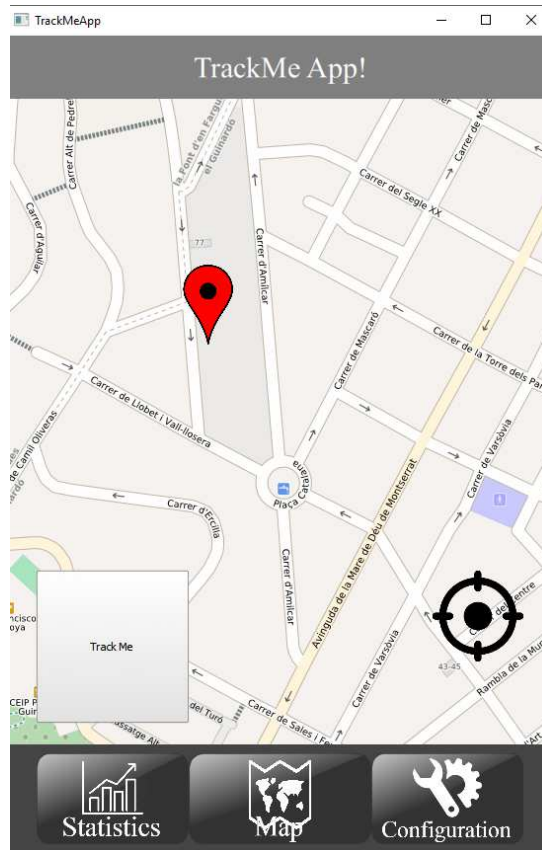


Figura 11.8. Finestra principal.

11.3.2.4. Pantalla d'estadístiques

Aquesta pantalla és per a fer alguns càlculs amb les dades rebudes del servidor. Per ara es farà una columna amb una estructura poc rebuscada però amb un toc de subtileza inspirada amb el menú d'ajustos d'android 4.4.

Cada element que hi ha a la finestra es troba dins de la columna a excepció del títol. A més a més s'hi afegeix un element per a poder abaixar la pantalla amb un moviment vertical amb la comanda:

```
Flickable {
    width: pagina0.width
    height: pagina0.height
    id: flickable
    contentHeight: ultima.y+basebotons.height + 150
```

Les dimensions seran les de la pantalla, i contentHeight descriu fins on baixarà la pantalla, en aquest cas fins a l'altura de l'element marcat com a ultima, que és l'últim rectangle de altura molt poca que simula una línia, més l'altura dels botons més un marge de 150 píxels.

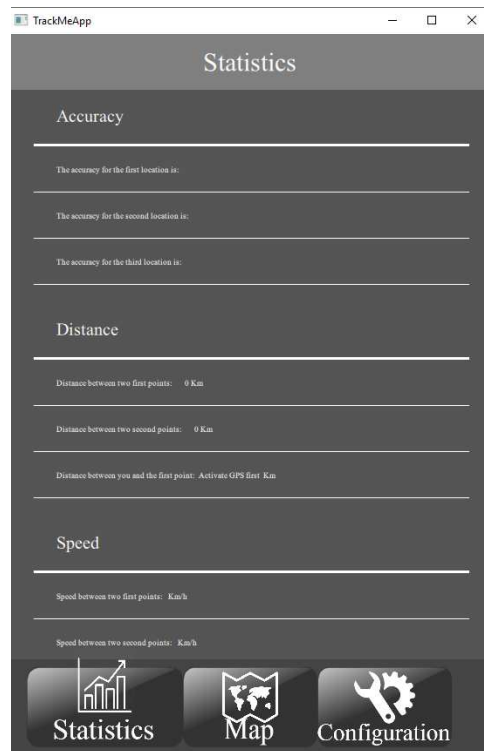


Figura 11.9. Finestra d'estadístiques.

11.3.2.5. Finestra de configuració

La finestra de configuració té opcions addicionals que en el seu moment permetran realitzar funcions extremes com la de mostrar la pròpia localització del dispositiu mòbil, no actualitzar dades del servidor o poder sortir de la aplicació.

L'estructura i el disseny és molt similar a la d'estadístiques, amb un fons fosc i línies i text blanc.

Els elements que s'incorporen en aquesta pàgina són interruptors. Per a col·locar-ne un es fa servir el següent codi:

```
Switch {
    x: showmygps.width + 20
    y: -10
    onChangedChanged: {}
}
```

El qual afegeix un interruptor clàssic en estat OFF. Quan es polsi, és a dir, quan canviï d'estat, tocarà programar-lo per a que faci el que es vulgui.

Per a sortir de la aplicació, s'ha de crear un botó que al ser premut faci un seguit d'operacions;

Haurà de retornar a la pantalla inicial, a més haurà de tornar a fer invisible tots els elements que ho eren en un principi, entre d'altres.

El resultat final queda així:

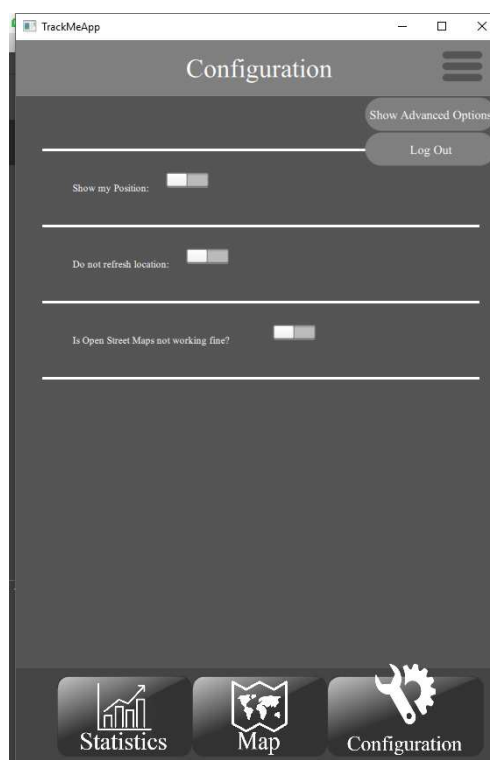


Figura 11.10. Finestra de configuració.

11.3.3. Prova de l'app a un dispositiu mòbil

Quan es prova tot el disseny en un mòbil per a veure com queda ja definitivament, hi ha diversos elements que canvien. Però abans d'entrar en detalls, hi ha un procediment per a pujar l'aplicació a qualsevol dispositiu mòbil.

Primer de tot, el mòbil ha d'estar en mode depuració per a poder córrer aplicacions d'aquest tipus baixades pel port microUSB. A més, la versió d'Android ha de ser superior a la que utilitza l'aplicació i que es defineix a Tools del programa Qt. Per últim, caldrà definir Android com a mètode de compilació del programa al crear aquest.

Fets aquests prerequisits, caldrà entrar a Build Settings de Android for armeabi, i estendre la pestanya que posa Build Android APK. Dins el menú desplegable s'haurà de prémer el botó Create Templates el qual crearà una nova carpeta dins del grup de carpetes de l'app anomenada Android.

Un dels problemes que s'ha trobat alhora d'executar l'aplicació pel mòbil, és que en aquesta carpeta d'Android, dins del codi de Android Manifest hi havia una propietat que no hi havia de ser i cada vegada que es crea un nou programa s'ha d'esborrar per a què funcioni. Aquesta és Layout Direction.

Amb tot això explicat, ja es pot compilar i executar l'aplicació en un mòbil. El procés tarda molt, uns quatre o cinc minuts i els resultats són els mostrats a la següent figura.

Si bé sembla que el resultat podria ser idèntic, no ho és. El botó és transparent mentre que al Qt no ho és, les lletres són diferents i al no poder saber les dimensions exactes de les paraules, no es pot centrar o ubicar en funció de la pantalla, i això portarà problemes de resolució, paraules més grans o desplaçades d'on originalment s'han pensat ubicar.

Tot i així el resultat és molt satisfactori, ja es pot visualitzar el progrés que es va fent en un dispositiu mòbil que serà per la plataforma a la qual anirà dissenyat tot el codi de l'aplicació.



Figura 11.11. Aplicació en un dispositiu mòbil HD de 5 polzades.

11.4. Programació de l'aplicació

Ja s'ha dissenyat l'aplicació i ara toca programar tots els elements perquè compleixin amb les funcions esperades: Al pitjar botons, connectar-se al servidor, que el mapa contingui elements dinàmics i que el Stack View funcioni com ho ha de fer. I es començarà amb això de fet.

11.4.1. Botons de canvi de pantalles

El Stack View s'ha de programar quan es premi algun dels botons que de canvi de pantalles juntament amb el de Sign In quan sigui correcte.

En el moment de registre, es crearà la condició que si l'usuari i contrasenya són els correctes, que es passi a la finestra principal, i que sinó, mostri el rectangle de torna-ho a intentar.

El codi simplificat és així:

```
onClicked: {  
    if (user.text== "VicençS" && password.text== "upc"){  
        rectgoodlogin.visible=true  
        miStackView.push(finestrappal)  
        finestrastadistic.visible=true  
        finestrappal.visible=true  
        root.finestra= 2  
    }  
    else {  
        rectfailedlogin.visible=true  
    }  
}
```

Si l'usuari i la contrasenya són els que s'observen, es donarà per vàlid i es mostrarà el missatge de confirmació amb un retard, i es faran, primer de tot un push de totes les finestres de l'aplicació per a que no doni cap error deixant com a última la principal que serà la que quedarà visible. Push és una característica del Stack View i posa per pantalla la finestra que se li declari. A més a més compte amb una animació com si estigués a la dreta de la pantalla anterior.

Ara es faran visibles les pantalles i els botons i es posarà una variable anomenada finestra cridada a l'arxiu main i per això dins de root, amb el valor 2. Aquesta variable indica en quina pantalla es troba l'aplicació en cada moment i es jugarà amb ella per a saber què fer amb les interaccions del Stack View.

Per acabar, si no és correcte la informació d'usuari i contrasenya, es farà visible el requadre de torna-ho a intentar, i justament el botó d'aquesta finestra anomenat try again, té la operació contrària, farà invisible el requadre.

Passem a veure com queda el codi de cada botó de les pantalles un cop registrat l'usuari.

```
onClicked: {
  if (finestra==1){
    miStackView.push(finestrappal)
  }
  else if (finestra==3){
    miStackView.pop(finestrappal)
  }
  finestra=2
}
```

Aquest codi és el del botó de la pantalla principal, situat al mig de les tres. Un cop clicat, la variable finestra passarà a valdre 2 (pensant-ho d'esquerra a dreta) i si la finestra valia 1, es farà un push (cap a la dreta) fins a la finestrappal. En canvi si val 3, s'haurà de fer un pop (l'acció contrària al push, tira enrere però cal tenir en compte que cal haver passat per ella abans o no funcionarà, d'aquí a que es fes un push a totes elles només registrar-se) que tindrà una animació cap a l'esquerra. Si la finestra valia 2, el programa no farà res ja que ja es trobarà dins la pantalla desitjada.

Hi ha hagut alguna complicació alhora de fer un pop de la finestra 3 a la 1, però seguint l'explicació es pot entendre la resta de botons. Es pot trobar el codi a l'Annex pel que no es detallarà més.

En el moment en què es vulgui sortir de l'aplicació, s'haurà de fer un pop a la finestra de registre. Aquesta opció es troba a configuració, dins la icona de opcions de dalt a la dreta amb el nom de log out. El codi simplificat és d'aquest estil:

```
onClicked: {
  desplegable.visible=false
  botomap.visible=false
  miStackView.pop(finestrainicial)
}
```

Primer caldrà amagar tot el que en un principi estava visible, falten moltes coses en el codi d'exemple però per a fer-se una idea, el desplegable del menú d'opcions de configuració, o els botons de canvi de pantalles ja que quan et registres no han d'estar disponibles. Per últim el pop a la finestra inicial.

11.4.2. Connexió amb el servidor

Connectar-se a internet no és tasca fàcil. Per sort existeixen programes d'exemple i les operacions de connexió són iguals. Cal dir que les dades no s'encripten en cap moment ni el servidor demana autenticació al entrar, això ho hagués complicat molt però sí que hi ha maneres per a fer-ho, i si es fes servir en algun moment dades bancàries o informació personal, caldria fer-ho.

En el moment que arriba la petició al ser pulsat el botó de Track me, l'aplicació comença un conjunt de passos fins a arribar en un moment que s'ha pogut connectar al servidor, no hi ha errors pel mig i les dades que arriben són correctes i ja es pot tractar la informació. Això ho fan dues funcions

auxiliars que es poden veure també al codi de la pàgina principal. En el moment en què la informació és fiable, s'ha de trossejar per a poder tractar-la. En informàtica es diu Parse i es fa amb JSon, un format de text per l'intercanvi de dades. En aquest moment s'ha de llegir diferents apartats de la matriu de files que es reben del servidor. L'estructura per a fer-ho és aquesta:

```
longitud0 = obj[0].longitud
```

On obj és una variable on hi ha afegit tot el text del servidor, [0] defineix a quina posició del vector es troba la informació, i .longitud defineix el nom del text que es vol guardar a la variable longitud0.

Seguint aquest exemple, s'ha de guardar la longitud, la latitud, el nombre de satèl·lits i l'hora la qual s'ha pres la informació. Queda així:

```
longitud0 = obj[0].longitud
latitud0 = obj[0].latitud
numsatellites0 = obj[0].numsatellites
var data0 = new Date(obj[0].tiempo)
date0= data0.getDate()
var month0=data0.getMonth()
var hours0=data0.getHours()
var minutes0=data0.getMinutes()
var seconds0=data0.getSeconds()
```

Per a poder interpretar correctament la data i hora, cal posar la informació en una variable de tipus Date. Si la declarem i se li demana .getMonth per exemple retornarà el mes al qual ha estat enregistrada. Això anirà bé per a calcular per exemple la velocitat entre dos punts de la matriu.

Al final s'ha seguit aquesta estructura per a enregistrar dins del mapa tres punts, els més recents però que alhora compleixin una condició:

- Dos punts hauran de tenir una separació mínima de 0.0005 graus de longitud i de latitud.

I si no la complís cap punt? Bé doncs també ha d'exceptuar la condició són les últimes coordenades de la llista rebuda del servidor.

Tot això es fa amb aquest bucle:

```
while (Math.abs(longitud1-longitud0) > 0.0005 && Math.abs(latitud1-
latitud0) > 0.0005 && x<47){
    x++
    temps1 = obj[x].tiempo
    longitud1 = obj[x].longitud
    latitud1 = obj[x].latitud
```

11.5. Ampliació de l'aplicació

Amb tota la informació rebuda del servidor, no només es pot mostrar al mapa tal i com s'ha fet al apartat anterior, també es pot i és necessari utilitzar-la per a realitzar totes les estadístiques que s'ha pensat fer en la seva finestra corresponent igual que les opcions extres de configuració.

11.5.1. Animacions

Per a donar-li una mica més de vida a l'aplicació s'ha dut a terme dues animacions en ella que involucren el fons de la pantalla inicial, que ara varia la seva propietat *x* per a què sembli que el mapa es mogui en un cicle tancat de uns segons i els tres botons principals.

```
SequentialAnimation on x {
    loops: Animation.Infinite
    NumberAnimation {
        from: -291; to: -550
        duration: 20000
    }
}
```

Loops significa el nombre de vegades que ha de fer l'animació, s'ha declarat com a infinit; Es declara que des de *x*=-291 fins a -550 (pel que anirà de dreta a esquerra), hi hagi una animació que duri vint segons. Un cop hagi arribat a -550, hi ha un altre NumberAnimation que retorna les *x* a la posició inicial per a completar el cicle.

L'altre animació dels botons engloba dues animacions en sí. Per una banda s'ha fet que la mida (width) dels botons augmenti si es troba la pantalla del botó, i l'altre fa que la posició *y* de la imatge que correspon al botó pugi alguns píxels.

Aquestes animacions tenen una manera de programar diferent que veurem a continuació.

```
PropertyAnimation { id: granst;
    target: botost
    property: "width"
    to: root.width/3.1
    duration: 200 }
```

Se li dona un nom a la animació d'una propietat, a quin element va dirigida la animació i quina propietat es modifica, i per últim quan ha de valdre quan passi la duració afegida. Per a activar l'animació, s'ha de fer dins d'un bucle com un if o un for, si es prem el botó d'estadístiques, `granst.running=true` és la línia de codi que activarà la animació.

Si es simula aquest codi per a tots els botons i després es fa amb les *y*, aconseguirem les animacions desitjades. Per descomptat cal fer una animació per a tornar enrere tant la amplada com la posició de les *y*.

11.5.2. Distància entre dos punts

Comencem doncs amb el tractament de les coordenades. Ens interessa saber la distància entre els últims dos punts localitzats per a posar un punt de partida. Per a fer-ho es necessitarà exclusivament les coordenades dels dos punts, però cal esmentar que la Terra no és esfèrica en la seva totalitat, i passar de coordenades decimals a metres varia en funció d'on ens trobem, ja que la circumferència de l'equador fa 40.075 Km i la dels pols 40.007 Km. Es podria estimar a:

$$\frac{40.075 + 40.007}{2} = 40.041 \text{ Km} \quad (\text{Eq. 11.1})$$

Si es pren aquest com el perímetre de la Terra, podem fer els següents càlculs:

$$\frac{40.041 \text{ Km}}{360^\circ} = 111,225 \text{ Km}/^\circ \quad (\text{Eq. 11.2})$$

Aproximant doncs d'aquesta manera es pot trobar que per cada grau, la distància que es recórrer són uns 111 Km.

La operació a fer finalment per a donar amb la solució amb les coordenades és la que s'expressa a continuació utilitzant Pitàgores.

$$|\text{Latitud1} - \text{Latitud2}| = \Delta\text{Latitud} \quad (\text{Eq. 11.3})$$

$$|\text{Longitud1} - \text{Longitud2}| = \Delta\text{Longitud} \quad (\text{Eq. 11.4})$$

$$D^\circ = \sqrt{\Delta\text{Latitud}^2 + \Delta\text{Longitud}^2} \quad (\text{Eq. 11.5})$$

$$D_{\text{Km}} = D^\circ \times \frac{111.225 \text{ Km}}{1^\circ} \quad (\text{Eq. 11.6})$$

D'aquesta forma ja podríem posar al codi aquesta operació i donaria un resultat aproximat de la distància en metres de dos punts en el mapa.

Per a fer-ho en el programa, s'han de fer algunes operacions per a que es faci correctament, al cap i a la fi, s'està programant amb Java Script.

```
inlongitud = Math.abs(longitud1-longitud0)
inlatitud = Math.abs(latitud1-latitud0)
root.dkm = Math.sqrt((Math.pow(inlongitud,2)+
Math.pow(inlatitud,2))*111225)
```

Creem dues variables que contindran el valor absolut (`Math.abs`) de la resta de les coordenades, s'anomenen `inclongitud` i `inclatitud`. L'arrel quadrada es representa amb `Math.sqrt` i elevar un valor es fa amb `Math.pow(base, exponent)`. Ara ja només farà falta posar a l'apartat d'estadístiques el següent:

```
text: qStr("Distance between two first points:" + root.dkm + " Km")
```

Per a que mostri el resultat a la aplicació.

Falta comprovar que els resultats siguin correctes ja que s'han comès aproximacions del radi de la Terra, parlem de valors molt grans que poden fer variar significativament els càlculs. A Google Maps, si es posen dues localitzacions que hagi detectat l'aplicació i que es trobin més o menys rectes, el resultat de la app i de el servidor de Google difereix de, més o menys, un quilòmetre. Això és degut a que a Espanya el radi mitjà que s'ha agafat no s'aproxima ben bé al real. Si s'agafa un més baix, en comptes de 40.041 Km, de 40.010 Km, el resultat ja dona molt més aproximat pel que aquest és el que finalment deixarem.

11.5.3. Nombre de satèl·lits

Vàrem deixar Arduino que ja enviava el nombre de satèl·lits als quals el receptor havia pogut obtenir dades i el servidor ho emmagatzemava correctament. És moment d'acabar aquesta tasca i mostrar-ho a l'aplicació. S'ha pensat que quants més satèl·lits ancorats, millor serà la precisió de les coordenades fent que si només hi ha tres o quatre, la precisió d'aquesta pugui variar de fins a 15 metres (almenys el que s'ha comprovat experimentalment). Per tant relacionarem el nombre de satèl·lits amb la precisió de la mesura.

Prenent una nova dada de la matriu de la base de dades obtinguda tal i com s'ha fet a l'apartat de connexió amb el servidor, es pot fer el següent codi:

```
if (numsatellites0<5){
    root.accuracy0 = "Limited"
}
else if (numsatellites0>=5 && numsatellites0<8){
    root.accuracy0 = "Optimal"
}
else root.accuracy0 = "Best"
```

D'aquesta manera es podrà utilitzar la variable `accuracy0`, que és un String, per a mostrar la qualitat de les coordenades.

```
text: qStr("The accuracy for the first location is: " +
root.accuracy0)
```


11.5.4. Velocitat entre dos punts

Es pot aprofitar la distància calculada anteriorment juntament amb la data a la qual s'han enregistrat les localitzacions per a trobar a quina velocitat aproximadament ha anat el dispositiu d'una posició a la següent.

El problema recau amb el tractament de la informació de la data. No és cap nombre ni es pot tractar com un conjunt de caràcters exactament. La solució està amb tractar-la com una variable de tipus Date. Aquesta és com un int o String però en JavaScript li pots demanar alguna dada com la hora, el dia, el temps en milisegons des de la mitjanit del dia 1 de gener de 1970... [79]

La funció de trobar els milisegons des de 1970 és molt bona idea per a utilitzar en el programa, però jo no he sigut capaç de que em retornés un resultat coherent, i després de una bona estona buscant quin era el problema s'ha trobat una altra alternativa.

```
var data0 = new Date(obj[0].tiempo)
root.date0= data0.getDate()
root.month0=data0.getMonth()
root.hours0=data0.getHours()
root.minutes0=data0.getMinutes()
```

Amb aquest conjunt de variables ja es podrà dir la diferència de temps entre dues localitzacions. Si ho pensem però hi haurà un moment el qual no es podrà predir la velocitat correctament i sortirà errònia. Aquest moment passarà cada 365 dies, exactament quan el month0 valgui 12 i el month1 valgui 01. Això resulta perquè en cap moment es demana l'any a l'aplicació, però opino que és una tonteria afegir més variables per un moment tant remot que es pot solucionar d'una altra manera que es veurà més endavant.

Si trobem les hores des del dia 01 de gener i les restem amb les de la segona localització es tindrà les hores que han passat entre dos punts i ja es podrà declarar la velocitat amb Km/h. Els càlculs per a fer aquesta operació són els següents:

$$(month0 - 1) \times 30 \times 24 = \text{Hores del dia 1 de Gener al dia 1 del mes.} \quad (\text{Eq. 11.7})$$

$$(date0 - 1) \times 24 = \text{Hores del dia 1 del mes al dia del mes.} \quad (\text{Eq. 11.8})$$

$$hours0 = \text{Hores des de les 00:00 a la hora enregistrada} \quad (\text{Eq. 11.9})$$

$$\frac{minutes0}{60} = \text{El que falta per a ser una hora dels minuts enregistrats} \quad (\text{Eq. 11.10})$$

Si aconseguim sumar totes aquestes hores tindrem el que volem. Per a posar un exemple, avui dia 18/05/17 a les 17:14, apliquem les operacions anteriors i tenim que:

- Portem 2880 hores en mesos.
- 408 hores en dies aquest mes.
- 17 hores avui.
- 0.25 hores en minuts.
- En total, 3304.25 hores des del dia 1 de gener a les 00:00.

$$V_{0-1} = \frac{D_{Km1}}{\Delta Hores_{0-1}} \quad (\text{Eq. 11.11})$$

El codi per a fer aquestes fórmules és el següent:

```
var ara0=(month0*720)+( (date0-1)*24)+hours0+(minutes0/60)
var ara1=(month1*720)+( (date1-1)*24)+hours1+(minutes1/60)
root.velocitat01= Math.round(root.dkm/(ara0-ara1)*1000)/1000
if (velocitat01<0 || velocitat01>300){
    root.velocitat01= "indefinite"
```

A la quarta i cinquena fila es pot observar la operació per a trobar les dues velocitats. Es pot veure que la fórmula conté primer un Math.round que arrodoneix al resultat a zero decimals. L'adequat serien un o dos decimals, (de fet es pot fer) però realment aquesta velocitat és aproximada pel que tenir o no decimals no és un fet molt important.

Per últim s'ha afegit una condició que en el cas que una velocitat valgui menys de 0 (és a dir, sigui negativa) o més de 300, que la velocitat valgui indefinida ja que valors d'aquesta mida no es poden donar en cap cas lògic de velocitat en Km/h.

Per a mostrar-ho en la aplicació afegirem un text que contindrà aquesta línia.

```
text: qStr("Speed between two first points: "+ root.velocitat01
+ " Km/h")
```

Al final, degut a que la precisió era baixa, s'ha tingut en compte també el nombre de segons a la mesura de velocitat. Les equacions se'ls hi ha d'afegir un terme més simplement.

11.5.5. Simulació de la posició actual.

Bé, no fa falta dir que totes aquestes ampliacions són optatives per al disseny de l'aplicació, però donat que les característiques essencials ja funcionen, s'ha decidit fer l'aplicació més atractiva.

Dins la variable de tipus Date que s'ha utilitzat abans, hi ha una comanda que permet saber l'hora actual. Amb aquesta restada de la que s'ha pres a la última posició podrem saber quan de temps fa que s'ha pres. Si a més ho combinem amb la velocitat a la qual anava el dispositiu l'últim cop que s'ha pres dades (cosa que ara mateix no es pot fer ja que el dispositiu no envia dades de velocitat, però podem aproximar-ho a la velocitat trobada al apartat anterior), podrem trobar a quina distància pot estar el dispositiu de nosaltres ara mateix.

```
actual=new Date()
```

Amb aquesta variable, actual passarà a valdre el temps actual. Si cridem funcions com actual.month, trobarem el mes, el dia, el minut i també els segons.

Repetirem les equacions per a trobar el nombre d'hores passades des del principi d'aquest any, però en comptes de deixar-ho en hores, ho deixarem en segons pel que tots els nombres s'hauran de multiplicar per 3600. Així la operació que queda és aquesta:

```
var ara= (month*2592000)+( (date-1)*86400)+(hours*3600)+(minutes*60)+seconds
```

Si restem el temps en segons de la última posició amb la variable ara trobarem el temps que ha passat, més dues hores! Recordem que el temps del servidor està en horari internacional, i el dispositiu en horari de la península (Sí, a un altre zona amb una franja horària diferent, això no funcionaria bé, ja ho arreglarem) pel que se li haurà de restar dues hores o 7200 segons.

Amb el temps del interval, només falta operar amb la velocitat del apartat anterior segons la següent equació:

$$\Delta X = V \times \Delta t$$

On ΔX serà la distància que haurà recorregut el dispositiu des l'últim punt (X_0), V serà la velocitat trobada anteriorment i Δt el temps que acabem de trobar.

L'última part és com mostrar-ho al mapa. La idea és que en la posició última, si la distància és menor a 2400 i més gran de 0 (així s'evita que marqui alguna cosa si la zona horària és diferent), es faci visible si la opció a configuració està activada, un altre QuickItem que serà un rectangle arrodonit (una circumferència). Però fins ara no importava si ampliaves o reduïes el mapa que els marcadors no variaven la seva mida. En canvi ara sí que ho ha de fer. Per a aconseguir-ho s'afegeix el següent al QuickItem:

```
zoomLevel: 10
```

Així el marcador estarà a la superfície del mapa amb una ampliació de 10 (pot ser qualsevol).

La solució que s'ha optat per a que la mida de la circumferència vagi variant correctament ha estat la de mesurar a Google Maps quan hi havia exactament entre dos punts d'un carrer recte. A prova i error s'ha anat provat diferents valors de mida amb el centre centrat al mig del carrer i s'ha arribat a que amb un zoomLevel de 10, els 130 metres del carrer eren un diàmetre de la circumferència de 0.7. No és la solució més acurada, però en cap lloc s'especificava res del zoomLevel dels objectes de qml.

Si mirem quan deixa de mostrar la circumferència, s'ha dit que quan el diàmetre valgui 2400, es farà invisible la circumferència. Això són uns 400 Km, pel que de radi seran 200 Km. A partir d'aquest radi ja no tindrà massa sentit seguir simulant la zona a la qual pot estar el dispositiu.

Per altra banda, el simulador no té en compte altituds ni parades, ni corbes, ni acceleracions ni frenades. Per a fer-ho més real, s'afegirà una disminució de la velocitat en un 40% pel que quedarà en un 60% de la velocitat total i que a més distància recorreguda, acabarà essent un 40% d'aquesta.

```
var tantpercent = 0.6 - 0.000004*distancia //Y=b-mX
if (tantpercent<0.4) { tantpercent=0.4 }
diameter = distancia*0.7/130*tantpercent
```

Es té present que no és la millor funcionalitat de l'aplicació, però sempre es podria millorar obtenint la velocitat del dispositiu i no el càlcul entre les últimes coordenades. Tot i així funciona força bé si l'última dada s'ha enregistrat fa poc i la velocitat pren un nombre lògic.



Figura 11.12. Mapa amb esfera de simulació engrandint-se.

A la imatge es pot veure que amb els canvis fets, a 70 Km/h i amb una hora i 10 minuts, el dispositiu pot trobar-se a Barcelona essent el origen Vilafranca del Penedès.

11.5.6. Localització GPS del mòbil

A configuració hi ha un botó que al activar-se es fa aparèixer una icona al mapa d'on es troba actualment el dispositiu. Al canviar d'estat el Switch, una variable s'activa o es desactiva en funció del que valgui el Switch. A l'arxiu principal hi ha un codi per a fer servir el receptor GPS del mòbil.

```
PositionSource {  
    id: gpsmobil  
    updateInterval: 1000  
    active: true  
    onPositionChanged: {  
        var coord = gpsmobil.position.coordinate;
```

Aquesta funció que es refresca cada segon, detecta el canvi de posició del mòbil a partir del receptor, i si ha canviat d'un segon a l'altre, posa a la variable coord les coordenades del mòbil.

Per a agafar les coordenades, només caldrà referir-se a elles així:

```
lat = coord.latitude
```

Es pot trobar el codi complet al final, on també hi ha programat que en funció de com troba la posició el receptor (ja que si està amb wifi pot ser que trobi unes coordenades menys precises i si està amb el receptor ancorat, les dades seran més fiables) i les operacions per a posar a la part d'estadístiques de distància entre el mòbil i l'última dada rebuda.

Per altra banda, quan el Switch de configuració es trobi en ON, un QuickItem com els que ja s'han vist es farà visible amb les coordenades de la localització de la variable coord.

Per últim, fa falta donar-li permisos a l'aplicació per a què demani cert accés al mòbil. Per exemple quan es vol descarregar una app de missatgeria és normal que demani acceptar que l'aplicació tindrà accés a la llista de contacte, en aquest cas, al receptor GPS. Per a fer això, cal anar a la carpeta d'Android que es crea a la pestanya de Projects, Android for armeabi, Build Android APK, Create Templates. A Android Manifest, a l'espai de Permissions, caldrà afegir les opcions de:

Android.Permission.ACCESS_FINE_LOCATION

Android.Permission.ACCESS_COARSE_LOCATION

La primera donarà permís a la localització del receptor GPS i la segona per Wi-fi.

11.5.7. Buscador lliure de posició

I si tres coordenades no són suficients? I si voldries veure on es trobava el dispositiu fa unes hores i no pas on es troba ara? Bé fins ara només es veien tres posicions del servidor, però es pot buscar una alternativa per a que sigui l'usuari qui triï quines coordenades vol veure.

Per a fer aquesta ampliació, es deixarà veure al usuari la llista completa de dades que arriben del servidor i se li donarà un QuickItem per a que introdueixi les coordenades que vulgui al mapa.

Aquesta opció es troba a la finestra de configuració, a opcions avançades dins la icona de opcions a dalt a la dreta. Quan es pitja, surt un requadre de precaució que indica que aquesta opció mostrarà les dades rebudes del servidor, si es decideix continuar, ho farà visible, i si no és el cas, farà invisible aquest requadre de precaució i prou.

En el moment que s'accepti, apareixerà un nou requadre amb una matriu de 50 files i dos entrades de text i un botó per a cercar en el mapa les coordenades introduïdes manualment.

El resultat és el que es mostra a les dues imatges següents.

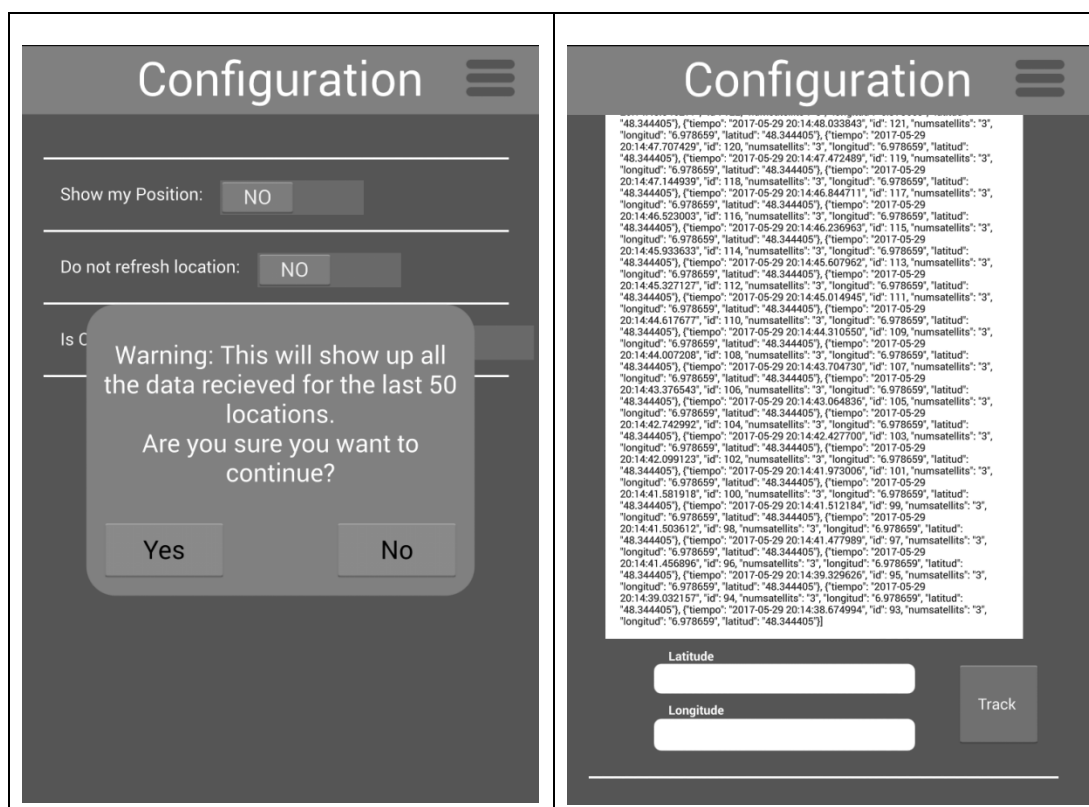


Figura 11.13. Opció de buscador lliure.

La línia de codi que permet mostrar tot el que es rep del servidor és aquesta:

```
text: qstr( root.obj )
```

Per a fer que entri en el rectangle blanc, cal posar la opció de wrapMode i les entrades de text són similars a les de registre. Quan es pitja el botó, agafa les coordenades i les posa a unes variables globals que es faran servir en un QuickItem i posa una variable booleana a True que farà la icona visible, però ha d'haver-hi unes condicions per a fer tot això; Que la latitud estigui compresa entre ± 83 i la longitud estigui entre ± 178 que són els valors que poden prendre com a coordenades. Si no és un nombre d'aquests o hi ha lletres o símbols, apareixerà un text informatiu en vermell conforme s'han introduït coordenades errònies.

Figura 11.14. Dades mal introduïdes.

11.5.8. Alternar mapes

L'última ampliació que es farà serà per a canviar de servidor de mapa. Algun cop (pocs de fet) s'ha observat com OSM no funciona massa bé i dona errors d'entrada als servidors que tenen allotjades les rajoles del mapa. S'ha provat amb altres servidors gratuïts, però definitivament OSM és la millor opció. Existeix un servidor anomenat HERE Maps que proporciona mapes utilitzables per mòbils però amb la obligació de registre i introducció de paràmetres. A més, aquests caduquen cada tres mesos i cal aleshores, pagar pel servei. Tot i que no és massa bona idea, funciona. S'ha registrat a Here Maps i amb la clau d'accés introduïda adequadament al codi, el mapa es veu correctament. Però a part d'algun problema de moviment al canviar entre mapes, no és una bona solució definitiva.

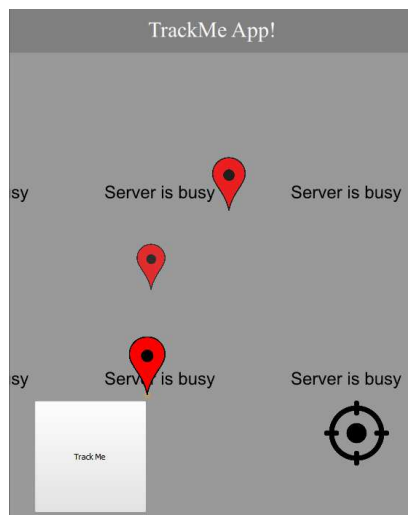


Figura 11.15. Error dels servidors de OSM.

S'ha acabat optant per a canviar entre mapes del propi servidor de OSM. Resulta que aquest servei ofereix diferents mapes (realment no es troben allotjats al mateix servidor, aquests altres mapes estan a thunderforest.com, però OSM n'és el contribuïdor pel que el seu accés es pot fer a través d'ell) amb diferents característiques. L'únic inconvenient és que tenen marca d'aigua si no estàs registrat a thunderforest.com. Al contrari de Here, per a aquests altres mapes no hi ha manera de posar la clau API correctament, almenys jo no n'he sigut capaç pel que segueixen amb la marca d'aigua.

Per a canviar amb aquests mapes, no es tracta de canviar el Plugin de OSM ja que també són d'aquesta companyia, es tracta de canviar el tipus de mapa, i l'única manera que s'ha aconseguit fer funcionar ha estat demanant a la funció de mapes suportables que posés el que es troba en la posició *i*.

M'explico després del codi:

```
onPluginChanged: {
  if (currentIndex+1 < map.supportedMapTypes.length) {
    currentIndex++
    map.activeMapType = map.supportedMapTypes[currentIndex]
  }
  else{
    currentIndex=0
    map.activeMapType = map.supportedMapTypes[currentIndex]
  }
}
```

Quan la variable plugin canviï (aquesta està enllaçada amb el botó de configuració), si la variable currntIndex és més petita que la longitud de la matriu de mapes suportables, incrementarà la

variable i passarà a estar actiu el mapa que suporti el servei a la posició que marqui la variable. Quan es trobi a l'últim mapa, la variable passarà a 0 pel que el mapa serà l'inicial de nou.

Pel que fa al botó de configuració, s'ha posat que tingui el text de la variable `currentIndex` per a què marqui en quin nombre de mapa es troba. El resultat és molt bo i dona validació a les hores dedicades en aquest apartat. A continuació es mostraran alguns exemples.

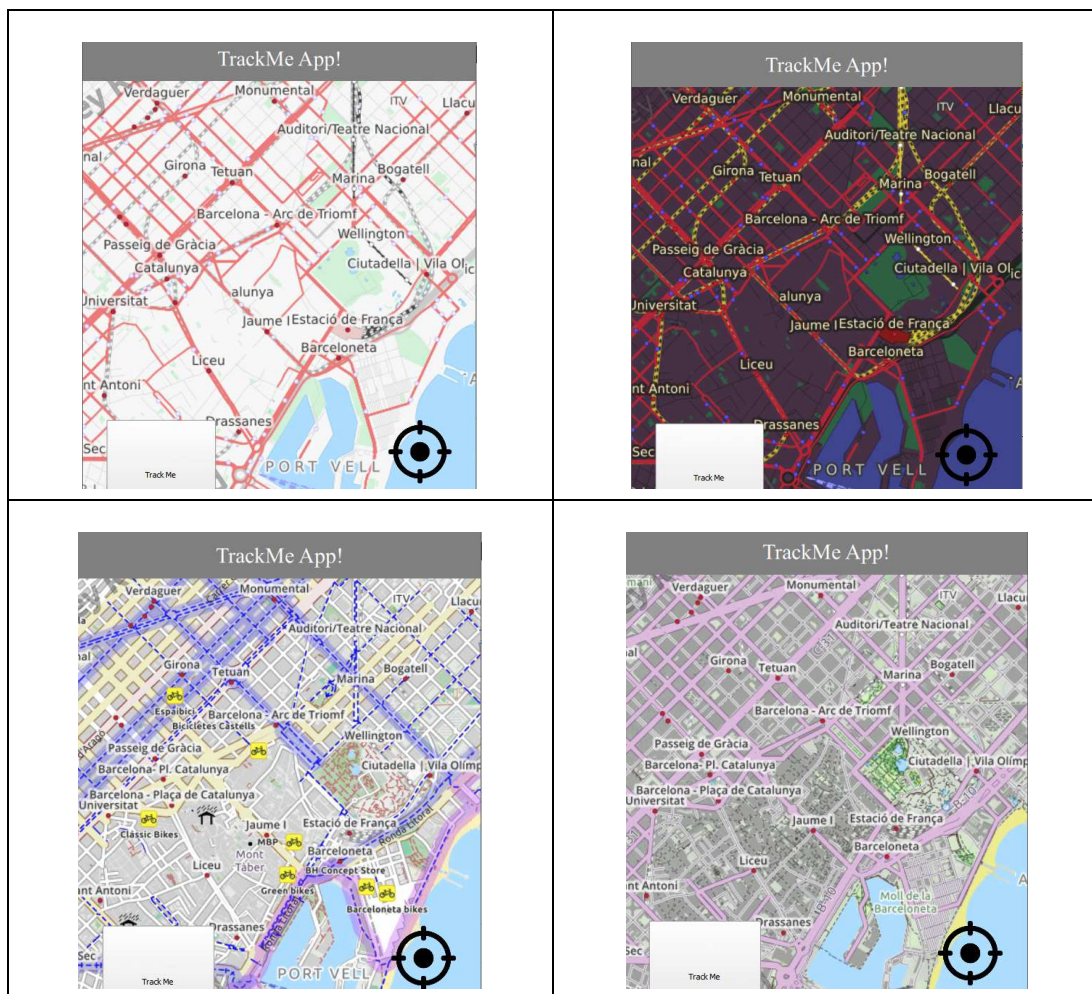


Figura 11.16. Diferents tipus de mapes.

12. Anàlisi de l'impacte ambiental

Per a la realització d'aquest projecte s'han utilitzat materials de terceres persones. El hardware en sí està fet principalment de silici i de plàstic, materials reciclables en la seva totalitat, però és difícil el seu aïllament. En el moment en que es vulgui llançar, s'ha de portar en un punt verd. Molts ajuntaments disposen d'aquest servei. També es poden portar a una planta de tractament de residus, però és més habitual la primera opció.

En el seu funcionament normal, l'impacte ambiental que genera el dispositiu és, pràcticament, nul. Funciona a partir d'energia elèctrica que no genera cap residu. L'única contaminació que es pot considerar és en forma d'ones electromagnètiques, que poden interferir en altres aparells electrònics.

Pel que fa a les bateries, aquestes estan fetes d'ió de Liti o Li-ió, que fan servir com a electròlit una sal de liti. Un mal ús d'aquestes o un cop fort pot provocar que la bateria rebi danys interns i s'incendiï o fins i tot, faci explosió.

Cal mantenir les bateries lluny de materials metàl·lics i conductors, no sotmetre-la a altes temperatures ni humitats extremes, i que en qualsevol senyal d'inflat o dany superficial, es reciclin i canvïin.

Per al reciclatge de les bateries cal portar-les a un punt net. Els propis locals que venen dispositius electrònics tenen per llei admetre components per al seu reciclatge pel que no és molt difícil trobar un punt en qualsevol ciutat o poble.

Conclusions

L'objectiu d'aquestes conclusions és el de ser el màxim objectiu i transparent possible en quant als resultats del projecte.

Si ens fixem amb l'abast que es va fer al principi, tot ell ha estat complert i ha sobrat temps per a fer alguna de les ampliacions que es comenten després. Un dispositiu adquireix coordenades GPS i les envia utilitzant una xarxa de llarg abast, un servidor les guarda i una aplicació les demana i les mostra en un mapa.

El treball que s'ha fet amb el microprocessador ha acabat essent molt més robust del que en un principi es pensava que seria. El problema més important al que s'enfrontava el codi funcional d'Arduino era que no funcionava si les coordenades tenien un sol caràcter més al esperat, per exemple, un negatiu. Per sort es va trobar la manera de solucionar-ho i ara encara que les coordenades siguin negatives, el projecte segurament (no s'ha provat, no m'he desplaçat fins a una zona amb aquestes condicions) segueixi funcionant correctament. Per altra banda, també s'ha incorporat la característica del nombre de satèl·lits, un afegit addicional i ja és impossible que les coordenades del GPS siguin 00.0000 degut a la no connexió del receptor GPS.

Com a punt parcialment negatiu, el projecte no pot funcionar a tot el món principalment perquè la targeta SIM amb la qual el mòdul es connecta a internet mitjançant el protocol TCP és de Symio i utilitza les antenes de telecomunicacions de la operadora Orange. Al país que no hi hagi Symio, no funcionarà el projecte. Però això no és motiu per a treure diferents versions amb SIM del país al qual vagi enfocat amb un simple retoc del codi d'Arduino. La resta funcionaria tot igual. S'hagués pogut posar un teclat alfanumèric per a poder fer d'aquest projecte un sense necessitat de retocar el codi base en cas de canviar la targeta SIM, o canviar d'operadora.

Com a ampliacions, l'aplicació mòbil és qui més en contempla. Té un disseny agradable i senzill, es simula una zona on podria trobar-se el dispositiu a temps real, es mostra la posició del mòbil i un seguit d'estadístiques, etc. La plataforma que menys millores ha rebut ha estat el servidor, que només guarda dades, permet a l'aplicació agafar-ne 50 i quan la BD està plena la reinicia. Arduino es va deixar acabat en un punt on donava algun error de tant en tant, es quedava congelat si no feia bé alguna comanda, ni tampoc hivernava, esperava coordenades correctes o funcionava si aquestes eren negatives. Tot això ja s'ha solucionat i de moment, encara no ha fallat més que quan no troba senyal GPS o GPRS, això ja són forces superiors; Tot i així en el moment que torna la senyal, torna a funcionar com s'esperava.

Tampoc s'ha incorporat cap manera de comunicar-se amb el dispositiu. Una ampliació molt atractiva que contemplava afegir un micròfon i altaveu que degut a un seguit de dificultats es va acabar per no dur a terme. Tampoc un giroscopi o acceleròmetre per a mesurar altres variables d'interès tot i que en el seu moment es va adquirir un giroscopi, però al final no va donar temps a provar.

En el cas que s'hagués posat un acceleròmetre, es podria haver millorat la zona de probabilitat de l'aplicació fent servir la velocitat que enviava el dispositiu i moltes altres millores referents a la informació rebuda dels sensors.

Bé, ja es veu que les ampliacions que se li poden fer a aquest projecte són moltes tant per software com per hardware però la idea principal del projecte s'ha desenvolupat correctament pel que podem concloure els resultats d'aquest com de l'objectiu del propi TFG (marcat com a aprenentatge autònom) com un èxit.

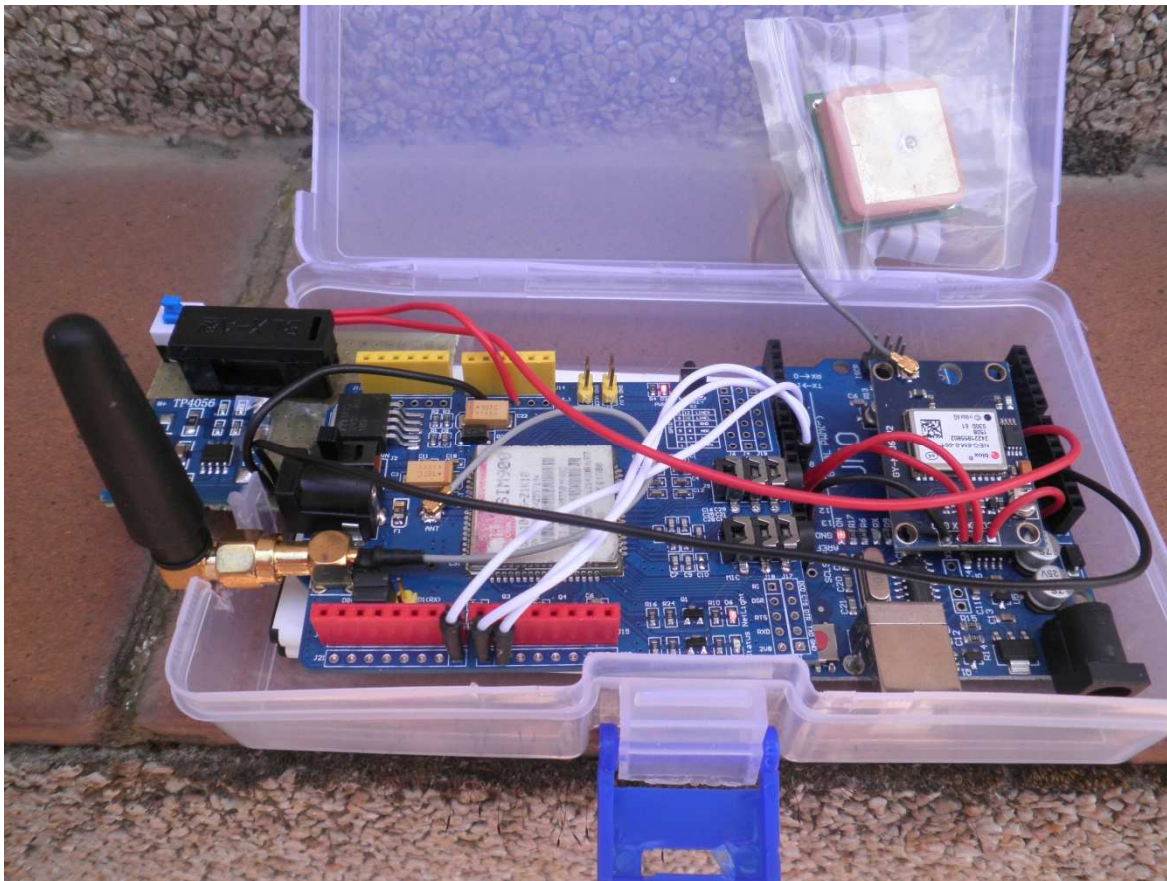


Figura C.1. Resultat final del prototip.

Pressupost

El pressupost es troba desglossat en quatre apartats: costos d'enginyeria, mà d'obra, materials i cost total del projecte.

Costos d'enginyeria

Pressupost referent a les hores dedicades al disseny del prototip (incloent cerca d'alternatives i millores), al disseny de l'aplicació i a tot el codi en general i als plànols o esquemes electrònics.

Especificació	Hores dedicades (h)	Preu (€/h)	Cost Total (€)
Disseny del prototip hardware	45h	20€/h	900€
Disseny del codi pel microprocessador	40h	20€/h	800€
Disseny del codi pel servidor	40h	20€/h	800€
Disseny del codi per l'aplicació	200h	20€/h	4000€
Plànols	4h	20€/h	80€
Total	329h		6580€

Taula 1. Pressupost dels costos d'enginyeria.

Mà d'obra

Pressupost referent a totes les hores dedicades a la construcció del prototip, a la redacció de la memòria i els seus annexes.

Especificació	Hores dedicades (h)	Preu (€/h)	Cost Total (€)
Muntatge del receptor GPS	2h	14€/h	28€
Muntatge del mòdul GPRS	5h	14€/h	70€
Muntatge bateria i mòdul de carga	7h	14€/h	98€
Redacció memòria	230 h	8€/h	1840€
Assemblatge i soldatge	8h	14€/h	112€
Total	252h		2148€

Taula 2. Pressupost dels costos per ma d'obra.

Materials

Pressupost referent als materials emprats per a la fabricació i disseny del prototip.

Especificació	Quantitat	Preu Unitat (€)	Cost Total (€)
Microprocessador Arduino UNO	1	7.27€	7.27€
Receptor GPS i antena	1	14.89€	14.89€
Mòdul SIM900	1	24.99€	24.99€
Targeta SIM Symio	1	7.0€	7.0€
Bateria 3000 mA/h	2	7.40€	14.80€

Mòdul carga	1	1.45€	1.45€
Botó d'enclavament	1	0.41€	0.41€
Fusible	3	0.25€	0.75€
Porta fusible	1	1.05€	1.05€
Placa de topes	1	2.32€	2.32€
Bq Aquaris E5 HD (2a ma)	1	100€	100€
ASUS X75V Series (2a ma)	1	350€	350€
Total			524.93€

Taula 3. Pressupost dels costos de material.

Cost Total

El cost total del projecte surt amb el càlcul de tot el pressupost anterior incloent els impostos pertinents als costos d'enginyeria i a la ma d'obra.

Concepte	Cost Total (€)
Costos d'enginyeria	6580€
Ma d'obra	2148€
Materials	524.93€
Total brut	9252.93€
21% IVA	1832.88€
Total	11085.81€

Taula 4. Pressupost del cost total del projecte.

Proposta comercial

Si haguéssim de fer un preu de venda al públic pel dispositiu, quantes més unitats es produïssin, més econòmic resultaria. Anem a fer una proposta donant una producció inicial de 100, 500 i 1000 unitats.

El preu del producte ha de tenir el preu dels materials, aquest import és fixe per a tot els productes, i una proporció dels costos d'enginyeria i redacció de la memòria. Pel que fa al preu de la ma d'obra, si és donés el cas que es produís en massa, aquesta seria molt més àgil i eficaç ja que aquí només s'ha fabricat un prototip. En aquest cas doncs es redueix el temps de producció a un temps lògic per a la feina de muntatge i soldatge dels components i no el temps que ha portat muntar, soldar i pensar com posar-ho en el prototip. Per altra banda, en compres de material a l'engròs, s'acostuma a fer un descompte, s'aplicarà un 10% en base a descompte de material. Tampoc contarem els estris per a fer les proves.

Per a una proposta de 100 unitats, el preu de venda que hauria de tenir el producte és el següent:

Concepte	Cost Total (€)
Costos d'enginyeria	65.8€
Ma d'obra	6€
Memòria	18.4€
Materials	61.1€
Total brut	151.3€
21% IVA	18.94€
Total	170.24€

Taula 5. Proposta comercial per a 100 unitats.

Considero que el preu és molt elevat per a les prestacions que dona el producte. Per cent setanta euros tens al mercat Smartphones d'altres prestacions que fan més del que fa el producte dissenyat.

Anem a veure com queda el preu de venda amb una producció inicial de 500 unitats.

Concepte	Cost Total (€)
Costos d'enginyeria	13.16€
Ma d'obra	6€
Memòria	3.68€
Materials	61.1€
Total brut	83.94€
21% IVA	4.79€
Total	88.74€

Taula 6. Proposta comercial per a 500 unitats.

La reducció de preu és considerable, la meitat que amb una producció de 100. Per últim, anem a veure amb una de 1000 unitats.

Concepte	Cost Total (€)
Costos d'enginyeria	6.58€
Ma d'obra	6€
Memòria	1.84€
Materials	61.1€
Total brut	75.52€
21% IVA	3.0€
Total	78.52€

Taula 7. Proposta comercial per a 1000 unitats.

Bé, el preu per a doblar la quantitat de producció inicial de 500 a 1000 unitats, rebaixaria el preu deu euros. Definitivament no sortiria rentable fer una producció en torn a les 100 unitats, però segurament 1000 sigui un excés d'estoc que no es podria vendre. El punt òptim entre preu/producció es trobaria entre les 500 unitats amb les conclusions que s'han extret del apartat.

Cal remarcar que no s'ha tingut en compte el preu de la targeta SIM ni de la seva recàrrega. El preu que cobra Symio per la targeta és referent a les despeses d'enviament, en sí la targeta és gratuïta. La seva càrrega vindria a càrrec del comprador, tot i que es podria jugar amb que ells paguessin una cota al mes més elevada (Symio fa pagar tant sols 1euro al mes) i rebaixar el preu del producte.

Així doncs, amb aquesta idea en ment el preu final al que es vendria el producte al usuari seria el següent:

Concepte	Cost Total (€)
Preu del producte per a 500 unitats	88.74€
Reducció	28.74€
Tarifa mensual	5.75€
Total	60€ + 5.75€ mensuals

Taula 7. Preu de venda al públic final.

Caldria una producció inicial de 500 unitats, però el preu final seria força atractiu. Es necessitaria però que els compradors paguessin almenys cinc mesos del producte per a no tenir pèrdues. Tot i així, els que no arribessin a aquest període, es podrien compensar amb els ingressos extra per part dels que superessin aquest temps.

Opinió, valoració del projecte

Un treball final de grau no té opinió! Bé... la guia per a redactar-lo animaven que l'estudiant fos creatiu, a personalitzar l'essència de la memòria, a més, la opinió subjectiva del redactor sempre l'he trobat interessant en qualsevol lectura.

Durant tot el treball he intentat ser tant objectiu com he pogut, (disculpes si algun cop m'ha sortit opinió en la redacció) i ara per a contrastar amb els sis mesos (haurien de ser quatre però vaig començar el treball abans, ja sabia que sinó em faltaria temps) que la meua opinió ha estat encadenada, vull transmetre les sensacions que he anat tenint mentre feia el treball.

Recordo el primer dia que vaig pensar, *avui vaig a fer TFG!* No tenia ni idea per on començar. El tutor em va dir que ja redactaria la memòria més endavant, que comencés a mirar com fer el dispositiu. Però per a fer-ho, necessitava saber com estava el mercat, com es feien aquest tipus de projectes, i això era un propi apartat de la memòria. Després d'una setmana sense haver fet res vaig començar pel principi, per la memòria, i he seguit així fins ara, que ja l'acabo; Mentre feia una cosa, la redactava a la memòria i ja estava pensant amb què em tocava fer després per a començar a redactar i buscar informació, comprar algun material si feia falta, etc.

Cada vegada que alguna cosa funcionava, era una petita (o no tant petita) espurna d'alegria. El GPS no funcionava en un principi, era defectuós i vaig necessitar una devolució. El segon (jo pensant que devia ser jo qui no l'estava fent servir bé) va funcionar com esperava. Marcava la meua casa a Google Maps amb una precisió increïble, i tot i abaixant les persianes, es connectava i enviava les coordenades! Ni m'ho creia.

El GPRS m'ha complicat molt la vida, i de fet fins al final he anat retocant coses del seu codi. No sé perquè, però si no li poses un bon retard, o una operació rere altre, el mòdul deix de funcionar. Sembla que es connecti a la xarxa per art de màgia, a més, afegint el codi per a tractar el GPS, va deixar de funcionar, i no, no tenien res a veure, quan acabava un començava l'altre. Per sort, no era que no funcionés, sinó que no mostrava les dades que rebia del servidor, però m'era igual, l'important era que es connectés al servidor.

No enganyaré si vaig estar un mes per fer un 10% del projecte. Estava en la fase que el dispositiu funcionava i havia de pensar on enviar les dades per a què algú les emmagatzemés. No puc més que agrair-li al meu professor d'informàtica l'ajut amb la creació del servidor. Realment el més difícil del projecte és interconnectar les plataformes entre elles.

I el més bonic de tot el treball, l'aplicació mòbil. Sincerament, animo a tothom que hi entengui a que faci una aplicació mòbil, a mi m'ha encantat. Anar veient com va prenent forma, des d'un rectangle a una finestra, des d'un mapa a un seguit de imatges botons i marques que interaccionen amb el mapa i et permeten fer el que vulguis si ets capaç de tenir la idea i buscar-te la vida per programar-ho. Estic segur que no serà l'última app que faci.

Aquest treball ha estat la culminació dels meus estudis que realment esperava, la professionalitat amb la qual s'han anat fent les coses, amb unes pautes marcades, ni jo mateix sabia que era capaç de fer un projecte així pel que n'estic molt satisfet. Per cert, totes les paraules d'aquesta memòria (55.000), han estat escrites del meu teclat; I els treball que m'he mirat de guia han estat gairebé nuls. No és que ho sàpigues fer ja tot, però jo tenia les meves idees marcades, i en tot el treball sabia què havia de fer a continuació. La guia de la universitat sí que me l'he mirat és clar, i com fer un anàlisi del impacte ambiental o seguir les indicacions de la pauta de word que facilita la universitat.

Per a tots els que creguin que fonaments no hauria de trobar-se en el treball, recorda'ls-hi que la pròpia universitat basa la seva metodologia en un 80% de teoria o fonaments. Els temes eren totalment desconeguts per a mi i redactar fonaments m'ha ajudat a entendre els conceptes i a veure per on s'enfocaria el meu treball, on tindria més dificultats i així anar estructurant la meua idea per a confeccionar el treball més elaborat dels meus estudis. Com a dada addicional, el haver de buscar informació sobre la relativitat general per a entendre el funcionament dels satèl·lits, m'ha obert una porta a un món fascinant, el de la física moderna i teòrica. Mecànica quàntica, matèria i energia, antimatèria, matèria obscura... Estic totalment enganxat a saber més del nostre univers.

No vull estendre'm més amb aquest apartat, m'agradaria però que més gent de la universitat pogués optar per a fer un projecte similar al meu, almenys a mi m'ha agradat molt fer-lo. D'electrònica clàssica en sí ni ha poca, és veritat, però també ho és que aquest projecte anava centrat en la interpretació de sensors i programació en general. Avui en dia està molt de moda això de rebre dades de sensors i emmagatzemar-les (Big Data?) i quin electrònic que s'apreciï no sap programar? Quantes empreses busquen electrònics i informàtics? Bé, almenys aquest és el meu punt de vista i pel qual no em penedeixo d'haver escollit el treball.

Bibliografia

Referències bibliogràfiques

- [1] Wikipedia Commons, "Global Positioning System," *Novembre 2016*. [Online]. Available: https://en.wikipedia.org/wiki/Global_Positioning_System. [Accessed: 17-Dec-2016].
- [2] Wikipedia Commons, "Satellite navigation Wikipedia," *Desembre 2016*. [Online]. Available: https://en.wikipedia.org/wiki/Satellite_navigation. [Accessed: 17-Dec-2016].
- [3] Wikipedia Commons, "Trilateration - Wikipedia," *Novembre 2016*. [Online]. Available: <https://en.wikipedia.org/wiki/Trilateration>. [Accessed: 17-Dec-2016].
- [4] David Pérez, "Todo sobre el GPS en Android," *Octubre 2015*. [Online]. Available: <http://www.elandroidelibre.com/2015/10/todo-sobre-el-gps-en-android-como-funciona-y-como-desactivarlo.html>. [Accessed: 16-Dec-2016].
- [5] TED-Ed, *How does your smartphone know your location? - Wilton L. Virgo - YouTube*. .
- [6] Richard W. Pogge, "GPS and Relativity," *28 Octubre 2016*. [Online]. Available: <http://www.astronomy.ohio-state.edu/~pogge/Ast162/Unit5/gps.html>. [Accessed: 19-Dec-2016].
- [7] Shane L. Larson, "GPS | Write Science," *Febrer 2015*. [Online]. Available: <https://writescience.wordpress.com/tag/gps/>. [Accessed: 18-Dec-2016].
- [8] Adam Davis, "How many GPS channels make sense? - Electrical Engineering Stack Exchange," *Maig 2011*. [Online]. Available: <http://electronics.stackexchange.com/questions/11884/how-many-gps-channels-make-sense>. [Accessed: 19-Dec-2016].
- [9] Ublox, "GPS Antennas."
- [10] Jeff Wilson, "Adding A GPS Chipset To Your Next Design Is Easy," *Agost 2011*. [Online]. Available: <http://electronicdesign.com/communications/adding-gps-chipset-your-next-design-easy>. [Accessed: 25-Dec-2016].
- [11] Bluetooth SIG, "Bluetooth® 5 Quadruples Range, Doubles Speed, Increases Data Broadcasting Capacity by 800% | Bluetooth Technology Website," 2017.
- [12] ISC JOSUE CARREÑO CAMACHO, "REDES INALAMBRICAS TESCO ISC." [Online]. Available: <http://redesinalambricasc.blogspot.com.es/2011/02/redes-inalambricas-de-area.html>. [Accessed: 28-Dec-2016].
- [13] Wikipedia Commons, "LMDS - Wikipedia." [Online]. Available: <https://es.wikipedia.org/wiki/LMDS>. [Accessed: 24-Dec-2016].

- [14] “Modulación de fase - EcuRed.” [Online]. Available: https://www.ecured.cu/Modulación_de_fase. [Accessed: 28-Dec-2016].
- [15] Wikipedia Commons, “Modulation Wikipedia,” *Octubre 2016*. [Online]. Available: <https://en.wikipedia.org/wiki/Modulation>. [Accessed: 28-Dec-2016].
- [16] Wikibooks Creative Commons, “Planificación y Administración de Redes,” *Març 2016*. [Online]. Available: https://es.wikibooks.org/wiki/Planificación_y_Administración_de_Redets/Tema_3/Datos_y_codificaciones. [Accessed: 29-Dec-2016].
- [17] Edumedia, *Demodulación de Amplitud – Vídeo – eduMedia*. .
- [18] Oscar M. Santa Cruz, “Cap.8.-RECEPCIÓN DE MODULACIÓN ANGULAR,” 2010.
- [19] “Diferencias entre GSM, GPRS y 3G,” *Maig 2011*. [Online]. Available: <https://www.taringa.net/post/ciencia-educacion/10530547/Diferencias-entre-GSM-GPRS-y-3G.html>. [Accessed: 29-Dec-2016].
- [20] Pive, “¿WiMAX o LTE? | Artículos,” *Diciembre 2014*. [Online]. Available: http://www.mibqyyo.com/articulos/2014/12/30/lte-o-wimax/#/vanilla/discussion/embed/?vanilla_discussion_id=0. [Accessed: 04-Jan-2017].
- [21] Guillermo Mayoraz, “Diferencias entre 2G, 2.5G, 3G, 3.5G y 4G,” *Diciembre 2010*. [Online]. Available: <http://tecnovortex.com/diferencias-entre-2g-2-5g-3g-3-5g-y-4g/>. [Accessed: 02-Jan-2017].
- [22] Wikipedia Commons, “4G Wikipèdia,” *Gener 2017*. [Online]. Available: <https://en.wikipedia.org/wiki/4G>. [Accessed: 03-Jan-2017].
- [23] “AT&T comenzará a probar la red 5G este año,” *Febrer 2016*. [Online]. Available: <http://expansion.mx/tecnologia/2016/02/15/att-comenzara-a-probar-la-red-5g-este-ano>. [Accessed: 03-Jan-2017].
- [24] “About Us | Google.” [Online]. Available: <https://www.google.com/about/behind-the-scenes/youtube-video/>. [Accessed: 30-Mar-2017].
- [25] Google, “Centros de datos de Google.” [Online]. Available: <https://www.google.com/about/datacenters/>. [Accessed: 25-Mar-2017].
- [26] Daniel Lopez, “Protocolos de Comunicación TCP/IP,” *Maig 2014*. [Online]. Available: <https://www.youtube.com/watch?v=Aie6av7qW9o>. [Accessed: 26-Mar-2017].
- [27] Wikipedia, “Familia de protocolos de Internet.” [Online]. Available: https://es.wikipedia.org/wiki/Familia_de_protocolos_de_Internet. [Accessed: 10-Mar-2017].
- [28] “TCP/IP.” [Online]. Available: <http://ccm.net/contents/292-tcp-ip>. [Accessed: 27-Mar-2017].
- [29] Julio César Chavez Urrea, “Protocolo TCP/IP,” 2010. [Online]. Available: <http://www.monografias.com/trabajos/protocolotcpip/protocolotcpip.shtml>. [Accessed: 27-

May-2017].

- [30] Kerrerk SB, "TCP vs UDP - What is a TCP connection? - Stack Overflow," *Novembre 2016*. [Online]. Available: <https://stackoverflow.com/questions/8156254/tcp-vs-udp-what-is-a-tcp-connection>. [Accessed: 27-Mar-2017].
- [31] Margaret Rouse, "What is HTTP?" [Online]. Available: <http://searchwindevelopment.techtarget.com/definition/HTTP>. [Accessed: 02-Jun-2017].
- [32] Statista, "Global smartphones sales by operating system 2009-2016 | Statistic," *2016*. [Online]. Available: <https://www.statista.com/statistics/263445/global-smartphone-sales-by-operating-system-since-2009/>. [Accessed: 01-Apr-2017].
- [33] Android, "Android - Historia." [Online]. Available: https://www.android.com/intl/es_es/history/#/marshmallow. [Accessed: 02-Apr-2017].
- [34] Tecnósfera, "Novedades de Android 8.0 o Android O - Apps - Tecnología - ELTIEMPO.COM," *Març 2017*. [Online]. Available: <http://www.eltiempo.com/tecnosfera/apps/novedades-de-android-8-0-o-android-o-70330>. [Accessed: 03-Apr-2017].
- [35] Amazon, "2016 GPS Tracker pulsera de reloj para niños edad avanzada GPS Google Map pulsera de botón SOS GPS Personal Tracker Localizador GSM SPG T58 reloj para iOS Android: Amazon.es: Electrónica." [Online]. Available: https://www.amazon.es/2016-GPS-avanzada-Personal-Localizador-T58-reloj/dp/B01MYRC2P4/ref=sr_1_5?ie=UTF8&qid=1496268838&sr=8-5&keywords=GPS+Child+Tracker. [Accessed: 01-Mar-2017].
- [36] Snowfox, "Snowfox Guardian on the go." [Online]. Available: <http://www.snowfoxfamily.com/>. [Accessed: 01-Mar-2017].
- [37] Universal Hawk, "GPS Tracker Mini impermeable IP54 para niños mayores mascotas perros equipaje T8S: Amazon.es: Electrónica." [Online]. Available: https://www.amazon.es/impermeable-IP54-para-mascotas-equipaje-T8S/dp/B01IR5CHFK/ref=sr_1_3?ie=UTF8&qid=1496268838&sr=8-3&keywords=GPS+Child+Tracker. [Accessed: 01-Mar-2017].
- [38] "GPS Buying Guide - SparkFun Electronics." [Online]. Available: https://www.sparkfun.com/pages/GPS_Guide. [Accessed: 20-Dec-2016].
- [39] - DronEPIC Aviation & Aerospace, "What is DGPS Differential GPS? How does it work? DGPS PDF," *Desembre 2015*. [Online]. Available: <http://dronepic.co.in/blog/what-is-dgps-differential-gps-how-does-it-work-dgps-pdf.html>. [Accessed: 28-Dec-2016].
- [40] Creative commons, "Real Time Kinematic - Wikipedia," *Novembre 2016*. [Online]. Available: https://en.wikipedia.org/wiki/Real_Time_Kinematic. [Accessed: 30-Dec-2016].
- [41] Samuel Buttarelli, "Will 2G and 3G networks be switched off soon?," *Octubre 2015*. [Online]. Available: <http://www.commscope.com/Blog/Will-2G-and-3G-networks-be-switched-off-soon/>. [Accessed: 02-Jan-2017].

- [42] AT&T Intellectual Property, "Frequently Asked Questions Regarding 2G Sunset," 2016.
- [43] Qualcomm, "LTE Carrier Aggregation." [Online]. Available: <https://www.qualcomm.com/invention/technologies/lte/lte-carrier-aggregation>. [Accessed: 04-Jan-2017].
- [44] Nicolás Rivera, "Así es Carrier Aggregation: hasta 375 Mbps en tu móvil," *Març 2015*. [Online]. Available: <https://hipertextual.com/2015/03/carrier-aggregation>. [Accessed: 04-Jan-2017].
- [45] Vodafone, "Mejor red 4G, tarifas y cobertura." [Online]. Available: <https://www.vodafone.es/c/conocenos/es/superioridad-de-red/4g-velocidad-conexion-internet/>. [Accessed: 04-Jan-2017].
- [46] AT&T Intellectual Property, "El Proyecto AirGig de AT&T Labs," 2016. [Online]. Available: <https://www.att.com.mx/noticias/att-proyecto-airgig.html>. [Accessed: 10-Jan-2017].
- [47] Amazon, "Amazon Web Services." [Online]. Available: <https://aws.amazon.com/es/ec2/pricing/>. [Accessed: 04-Apr-2017].
- [48] Microsoft, "Microsoft Azure." [Online]. Available: <https://azure.microsoft.com/es-es/overview/what-is-azure/>. [Accessed: 04-Apr-2017].
- [49] Google, "App Engine Google Cloud Platform." [Online]. Available: https://cloud.google.com/appengine/?utm_source=google&utm_medium=cpc&utm_campaign=2017-q1-cloud-emea-gcp-bkws-freetrial&gclid=Cj0KEQjwil3HBRDv0q_qhqXZ-N4BEiQAOTiChknBY_4NgdeXD-5TBjnSYCr19fx6q4UCYTK-gKBIYMwaAt4K8P8HAQ. [Accessed: 04-Apr-2017].
- [50] Heroku, "Pricing | Heroku." [Online]. Available: <https://www.heroku.com/pricing>. [Accessed: 04-Apr-2017].
- [51] "World Versus - Android vs Apple iOS." [Online]. Available: <http://worldversus.com/Android-vs-AppleiOS>. [Accessed: 28-Mar-2017].
- [52] Lifewire, "Android OS or Apple iOS - Which Is Best for Developers?," *Maig 2011*. [Online]. Available: <https://www.lifewire.com/android-os-vs-apple-ios-for-developers-2373400>. [Accessed: 27-May-2017].
- [53] Javier Pastor, "Desarrollo de aplicaciones móviles (I): así está el mercado," *Març 2014*. [Online]. Available: <https://www.xatakamovil.com/mercado/desarrollo-de-aplicaciones-moviles-i-asi-esta-el-mercado>. [Accessed: 27-May-2017].
- [54] "Android Vs Apple Wallpaper Tease Your iOS Friend." [Online]. Available: <http://www.theandroid-mania.com/android-vs-apple-wallpaper-tease-ios-friend/>. [Accessed: 28-Mar-2017].
- [55] Microsoft, "Mobile App Development | Visual Studio." [Online]. Available: <https://www.visualstudio.com/es/vs/mobile-app-development/>. [Accessed: 17-Apr-2017].
- [56] KDevelop, "KDevelop IDE." [Online]. Available: <https://www.kdevelop.org/>. [Accessed: 17-

- Apr-2017].
- [57] Qt Creator, “Qt - Product | The IDE.” [Online]. Available: <https://www.qt.io/ide/>. [Accessed: 17-Apr-2017].
- [58] unity, “Unity - Game Engine.” [Online]. Available: <https://unity3d.com/es>. [Accessed: 17-Apr-2017].
- [59] Punto Flotante S.A., “MODULO-SKYLABSKM53-GPS.” [Online]. Available: <http://www.puntoflotante.net/MODULO-GPS-SKYLAB-SKM53-PARA-MICROCONTROLADORES.htm>. [Accessed: 26-Dec-2016].
- [60] u-blox, “NEO-6 series | u-blox,” 2017. [Online]. Available: <https://www.u-blox.com/en/product/neo-6-series>. [Accessed: 26-Dec-2016].
- [61] Cooking hacks, “Cooking haks ‘GPRS.’” [Online]. Available: <https://www.cooking-hacks.com/catalogsearch/result/?cat=&q=GPRS&order=pr&dir=desc&limit=64>. [Accessed: 18-Jan-2017].
- [62] Adafruit, “Adafruit Ultimate GPS Breakout.” [Online]. Available: <https://www.adafruit.com/product/746>. [Accessed: 27-Dec-2016].
- [63] “Shields: Arduino Store.” [Online]. Available: <https://store.arduino.cc/category/107>. [Accessed: 18-Jan-2017].
- [64] Cetronic, “Adafruit Modulo 3G, GSM y GPS.” [Online]. Available: http://www.cetronic.es/sqlcommerce/disenos/plantilla1/seccion/producto/DetalleProducto.jsp?idIdioma=&idTienda=93&codProducto=999334089&cPath=1342&gclid=CjwKEAiAtefDBRD TnbDnvM735xISJABlvGOvmhZYg4htxHlp5ZbWTf0v4uVnnPccqMTafxphHCsdSBoCKGzw_wcB. [Accessed: 18-Jan-2017].
- [65] Eric Gakstatter, “What Exactly Is GPS NMEA Data?,” *Febrer 2015*. [Online]. Available: <http://gpsworld.com/what-exactly-is-gps-nmea-data/>. [Accessed: 20-Jan-2017].
- [66] cnxsoft, “Ublox NEO-6M \$10 GPS Module with Antenna Works with Arduino, Raspberry Pi, etc.,” *Gener 2016*. [Online]. Available: <http://www.cnx-software.com/2016/01/11/ublox-neo-6m-10-gps-module-with-antenna-works-with-arduino-raspberry-pi-etc/>. [Accessed: 04-Feb-2017].
- [67] “u-blox 6 GPS Modules Data Sheet.”
- [68] Paul McWhorter, “LESSON 24: Understanding GPS NMEA Sentences | Technology Tutorials,” *Agost 2014*. [Online]. Available: <http://www.toptechboy.com/arduino/lesson-24-understanding-gps-nmea-sentences/>. [Accessed: 20-Jan-2017].
- [69] Mikal Hart, “TinyGPS++ | Arduiniana,” *Setembre 2014*. [Online]. Available: <http://arduiniana.org/libraries/tinygpsplus/>. [Accessed: 25-Jan-2017].
- [70] LessonStudio, “Easy GPS connection to an Arduino,” *Gener 2016*. [Online]. Available: <https://www.youtube.com/watch?v=09sNLmoJ3HI&t=28s>. [Accessed: 10-Mar-2017].

- [71] ElectroDragon, "SIMCOM AT Commands." [Online]. Available: http://www.electrodragon.com/w/SIMCOM_AT_Commands. [Accessed: 10-Feb-2017].
- [72] MICREL, "MIC29302BU Datasheet." [Online]. Available: <http://pdf1.alldatasheet.com/datasheet-pdf/view/74604/MICREL/MIC29302BU.html>. [Accessed: 28-May-2017].
- [73] SEMTECH, "SMF05C pdf." [Online]. Available: <http://pdf1.alldatasheet.com/datasheet-pdf/view/42602/SEMTECH/SMF05C.html>. [Accessed: 28-May-2017].
- [74] Prometec, "MÓDULO GSM/GPRS: llamar y enviar SMS." [Online]. Available: <http://www.prometec.net/gprs-llamar-enviar-sms/>.
- [75] Prometec, "Conectarse a Internet mediante GPRS." [Online]. Available: <http://www.prometec.net/gprs-internet/>.
- [76] Arduino Playground, "ArduinoReset." [Online]. Available: <http://playground.arduino.cc/Main/ArduinoReset>. [Accessed: 02-May-2017].
- [77] Arduino Playground, "ArduinoSleepCode," 2006. [Online]. Available: <http://playground.arduino.cc/Learning/ArduinoSleepCode>. [Accessed: 02-May-2017].
- [78] QtStudios, "QtWS15- Developing with Qt Location, Laszlo Agocs, The Qt Company - YouTube," *Octubre 2015*. [Online]. Available: <https://www.youtube.com/watch?v=2u5wnrx6J-E>. [Accessed: 01-May-2017].
- [79] w3schools, "JavaScript Date Reference." [Online]. Available: https://www.w3schools.com/jsref/jsref_obj_date.asp. [Accessed: 18-May-2017].

Bibliografia complementària

- [1] Matjaz Vidmar, “A homemade receiver for GPS & GLONASS satellites”. [Online]. Available: <http://lea.hamradio.si/~s53mv/navsats/theory.html>

Quan s’ estava pensant com obtenir senyal GPS, es va trobar aquest document que ensenya com s’ha fet un receptor GPS casolà. El projecte és molt extens i no és una meravella, aquest va ser el motiu principal per a indagar més sobre els receptor comercials.

- [2] Cooking Hacks, “3G/GPRS shield over Arduino and Raspberry Pi”. [Online]. Available: <https://www.cooking-hacks.com/documentation/tutorials/3g-gps-shield-arduino-raspberry-pi-tutorial/>

Aquest és un tutorial molt complet sobre un mòdul 3G/GPRS que mostra pas per pas com fer funcionar el mòdul d’aquesta companyia. Si s’haugés obtingut aquest model, de ben segur s’hagués fet servir com ajuda. Malauradament el preu i el temps no han fet possible el pas a la tercera generació. També tenen per alla un mòdul 4G. Comprar-lo és impensable pel preu.

- [3] Electrodragon, “SIMCOM AT Commands”. [Online]. Available: http://www.electrodragon.com/w/SIMCOM_AT_Commands

Pàgina on tenen molta informació de les comandes AT existents. Interessant per a veure tot el que el mòdul GPRS és capaç de fer.

- [4] Linus Tech Tips, “GPS collar for my kids”. [Online]. Available: <https://www.youtube.com/watch?v=VOkktkibRbQ>

Vídeo que opina sobre un dels models de mercat similars al produït en el projecte. El dilema principal és si els nens haurien de portar un “collar” GPS per a estar localitzats les 24h del dia.



